University of Zurich[UZH]

Hypothesis-Based Collaborative Filtering

Retrieving Like-Minded Individuals Based on the Comparison of Hypothesized Preferences

A dissertation submitted to the
Faculty of Economics, Business Administration and Information Technology
of the University of Zurich

for the degree of
Doctor of Science

by
Amancio Bouza
from Spain

Accepted on the Recommendation of
Prof. Abraham Bernstein, Ph. D.
Prof. Dr. Harald C. Gall

2012

Amancio Bouza
http://www.cpoet.net

Photo Credits
Cover page (front): "Going Through A Tunnel With High Speed"
©iStockphoto.com/treekid (#17978622)

ISBN: 978-1-105-58508-1

The Faculty of Economics, Business Administration and Information Technology of the University of Zurich herewith permits the publication of the aforementioned dissertation without expressing any opinion on the views contained therein.

Zurich, March 2012

Head of the Ph.D. program: Prof. Abraham Bernstein, Ph.D.

Acknowledgements

It is my pleasure to thank the people who accompanied me during my dissertation and who made this thesis possible.

First of all, I would like to express my sincerest gratitude to my two advisors, Abraham (Avi) Bernstein and Harald Gall, for welcoming me in both research groups, for their company, for their feedback and for their support. Thank you Avi and Harald for providing me the freedom of developing my own research ideas and for being patient. Especially, thank you Harald for the infinite Coffee supply and the great time.

I would like to thank my colleagues of the Software Evolution and Architecture Lab (s.e.a.l.) group and the Dynamic Distributed Information Systems (DDIS) group for procrastinating when there was (or not) a need, chit chatting about life and research, and all the great and unforgettable time. Special thanks to the trio Flori, Jayalath, and Philip, as well as to Patrick, Gigs, Tom, Khoa, Patrick, Lorenz, Cosmin, Yi, Chaman, Michael, Giacomo, Matthias, and the crazy Swiss guy Sandro. Furthermore, I would like to thank all the students I supervised.

Special thanks to Patrick Knab and Emanuel (Gigs) Giger. I am grateful to Patrick for his steady and encouraging words as well as for all the long discussions about all the world and his wife, and about Golf, of course. I am grateful to Patrick for being such a great team player and providing good karma to the group. Many thanks to Emanuel (Gigs) Giger for sharing his deep knowledge about statistics and all the fun besides that.

I would like to thank the Swiss innovation promotion agency CTI, which partly supported this dissertation (KTI Nr. 8976.1 PFES-ES). In this context, I would like to thank Gerald Reif for his administrative support.

I am grateful to my parents, Maria and Antonio, for their love. Thank you mom, for your unconditional support during my entire life despite taking care of my dad so carefully for such a long time and still smiling. Mom, you are truly a heroine. I am grateful to still have you.

My deepest and sincere thanks go to Janette Garcia. I am grateful for your constant support, competent feedback regarding Economics, Mathematics and my thesis, for your endless patience when I am obsessed with my passions, for always believing in me, and for your love. Janette, I am grateful to have you.

— Amancio Bouza
University of Zurich, Switzerland, March 2012

To Janette and my Parents

Abstract

The vast product variety and product variation offered by online retailers provide an amazing amount of choice options to individuals, thus posing a big challenge to them finding and choosing interesting products which provide them the most utility. Consequently, consumers have to be satisfied with finding a product that provides them sufficient utility. Beyond that, individuals tend to even defer product choice [Dhar, 1997].

Recommender systems have emerged in the past years as an effective method to help individuals with finding interesting products. As a result, the consumer welfare enhanced by $731 million to $1.03 billion in the year 2000 due to the increased product variety of online bookstores [Brynjolfsson et al., 2003]. Consumer welfare refers to consumers' total satisfaction. This enhancement in consumer welfare is 7 to 10 times larger than the consumer welfare gain from increased competition and lower prices in the book market [Brynjolfsson and Smith, 2000]. In other words, recommender systems are essential for increasing consumers welfare, which ultimately leads to an increase of economic and social welfare.

Typically, recommender systems use the collective wisdom of individuals for exposing individuals to products which best fits their preferences, thus maximizing their utility. More precisely, the product ratings of like-minded individuals are considered by the recommender system to provide individuals recommendations. Commonly, like-minded individuals are retrieved by comparing their ratings for common rated products. This filtering technology is commonly referred to as *collaborative filtering*.

However, retrieving like-minded individuals based on their ratings for common rated products may be inappropriate because common rated prod-

ucts may not necessarily be a representative sample of two individuals' preferences being compared. There are four reasons. Firstly, the set of common rated products is too sparse to draw a significant conclusion about the preference similarity of both individuals.

Secondly, ratings for common rated products correspond to the intersection of two individuals' rated products and thus may represent only partially both individuals' preferences. Consequently, overall preference similarity is, in fact, deduced from partial preference similarity.

Thirdly, the preference similarity between two individuals is not assessable in the case when both individuals do not share ratings for the same products. Consequently, like-minded individuals are missed due to lack of ratings.

Lastly, retailers collect only a fraction of individuals' ratings on their store, because individuals purchase products from different stores. Hence, individuals' ratings are distributed across multiple retailers, which limits the set of common rated products per retailer.

In this dissertation, we propose *hypothesis-based collaborative filtering* (HCF) to expose individuals to products which best fits their preferences. In HCF, like-minded individuals are retrieved based on the similarity of their respective hypothesized preferences by means of machine learning algorithms hypothesizing individuals' preferences. Machine learning is a method to extract patterns to generalize from observations, thus being adequate to hypothesize individuals' preferences from their product ratings.

Generally, the similarity of two individuals' hypothesized preferences can be computed in two different ways. One way is to compare the hypothesized utilities which products provide to both individuals. To this goal, we use both individuals' hypothesized preferences to predict the utilities of some products. To compute the preference similarity, we propose three similarity metrics to compare product utilities.

The other way is to analyze the composition of both individuals' hypothesized preferences. For this purpose, we introduce the notion of *hypothesized partial preferences* (HPPs), which are self-contained and form the components which constitute hypothesized preferences. We propose several methods to compare HPPs to compute the similarity of two individuals' preferences.

We conduct a large empirical study on a quasi benchmark dataset and diverse variation of this dataset, which vary by means of sparsity degree, to evaluate the cold-start behavior of HCF. Based on this empirical study, we provide empirical evidence for the robustness of HCF against data sparsity and the superiority to state-of-the-art collaborative filtering methods.

We use the research methodology of grounded theory to scrutinize the empirical results to explain the cold-start behavior of HCF for retrieving like-minded individuals relative to other collaborative filtering methods. Based on this theory, we show that HCF is more efficient in retrieving like-minded individuals from large sets of individuals and is more appropriate for individuals who provide few provide ratings. We verify the validity of the grounded theory by means of an empirical study.

In conclusion, HCF provides individuals better recommendations, particularly for those who provide few ratings and for frequently rated products, which complicates the retrieval of like-minded individuals. Hence, HCF increases consumers welfare, which ultimately leads to an increase of economic and social welfare.

Contents

List of Figures

List of Tables

List of Algorithms

"So sehr ein Mann sich auch selbst empfiehlt, so sehr begünstigt die Empfehlung eines Freundes die ersten Augenblicke der Bekanntschaft."

—Johann Wolfgang von Goethe (1749–1832)
to Wilhelm von Humboldt, November 29th, 1801

I
Setting the Scene

1

Introduction

CONSUMERS' preferences and needs are individually different. For this reason, consumers receive different satisfaction from consuming a particular good because it fits consumers' preferences and needs to a different degree. For instance, a consumer who prefers red wine over white wine typically receives more satisfaction from enjoying a red wine relative to enjoying a white wine. As a result, consumers are willing to pay more for a good that better fits their preferences and needs. In other words, consumers act rational [Blume and Easley., 2008]. The highest price a consumer is willing to pay for a particular good is known as the consumer's reservation price.

Retailers exploit the heterogeneity in consumers' preferences by expanding their assortment, which better meets the divers preferences of consumers [Hoch et al., 1999]. The reason is that retailers can gain an economic advantage by skimming consumer surplus and increasing market share. Consumer surplus is the positive difference between the consumers' reservation price and the good's price. Retailers can exploit the higher reservation price of consumers due to the increased utility that a good provides to the appropriate consumers. More precisely, retailers demand a higher price for a good from consumers to skim their consumer surplus.

Market share can be increased relative to retailers that do not expand their assortment. The reasons are twofold. Firstly, consumers are more likely to find what they want in a wider assortment, thus consumers choosing retailers with wider assortments in the effort to reduce search costs [Hoch et al., 1999, Gourville and Soman, 2005, Kahn, 1995]. Secondly, consumers seek product variety due to satiation, curiosity or fluctuating requirements [Kahn, 1995]. Actually, consumers' utility can be increased solely by the fact of great product variety, a phenomenon known as the love of variety [Dixit and Stiglitz, 1977].

However, brick and mortar (B&M) retailers[1] are limited by physical constraints that are, specifically, the limited physical size of the store and the limited space on goods shelves to promote different products and amounts [Anderson, 2006]. Consequently, B&M retailers need to consider the opportunity costs of individual products and its amount. The opportunity costs is the loss of profit offering the second best alternative product in a certain amount. Hence, B&M retailers limit their offer to the most profitable products.

Advances in information technology and the Internet enables online retailers to vastly expand the variety and variation of products they provide to consumers [Hinz et al., 2011]. In contrast to B&M retailers, online retailers are not limited by physical constraints, thus leading to a dramatical increase in assortment sizes [Brynjolfsson et al., 2003]. Especially, the marginal costs of information goods (e.g. movies, music) are close to, if not exactly, zero [Smith et al., 2001] and do not require stock ground because they are infinitely reproducible on demand. Table 1.1, which is provided in [Brynjolfsson et al., 2003], shows the substantial differences in assortment sizes between the online retailer Amazon.com[2] and typical large B&M retailers. In the case of books, Amazon.com provides 2.3 million different

[1]The term of brick and mortar retailers refers to companies which use stores for operations; it is used to contrast to online retailers.

[2]`http://www.amazon.com`

books compared to 40 000–100 000 provided by typical large B&M retailers, which is 23 times more than Barnes & Noble[3] and 57 times more than typical large B&M retailer.

Product category	Amazon.com	B&M retailers
Books	2 300 000	40 000–100 000
CDs	250 000	5 000–15 000
DVDs	18 000	500–1 500
Digital cameras	213	36
Portable MP3 players	128	16
Flatbed Scanners	171	13

Table 1.1: Comparison of product variety between online typical large B&M retailers as presented in [Brynjolfsson et al., 2003].

However, the vast product variety and product variation pose a big challenge to consumers finding and choosing interesting products that provide them the most utility. Consequently, consumers have to be satisfied with finding a product that provides them sufficient utility. Beyond that, consumers tend to even defer product choice facing a vast product variety [Dhar, 1997]. The negative effect of vast product variety and product variation is called overchoice [Toffler, 1970]. For instance, Boatwright and Nunes reported that the revenue of an online grocery increased by 11% after decreasing the assortment across different product categories by 22% to 82% [Boatwright and Nunes, 2001].

Recommender systems drastically mitigate the effects of wide assortments [Hinz et al., 2011], namely information overflow in terms of vast product variety and overchoice. A recommender system is an information system that provides consumers recommendations regarding products which best fits their preferences, thus reducing information overflow and mitigating overchoice. In fact, recommender systems implement a distinct search

[3] `http://www.barnesandnoble.com`

paradigm relative to the canonical search paradigm of search engines that is: interesting products find the consumer instead of the consumer search for these.

Recommender systems use the collective wisdom of consumers for exposing consumers to products that best fits their preferences. This filtering technology is commonly referred to as *collaborative filtering*, a term coined by Goldberg et al. in [Goldberg et al., 1992]. These systems address two issues that search engines are lacking [Konstan, 2004]. Firstly, recommender systems can assess the quality of a particular good based on the collective wisdom of consumers by means of ratings. For this reason, recommender systems can predict more accurately the utility a good provides to consumers. As a result, recommender systems filter better the goods that provide the most utility to consumers.

Secondly, recommender systems are typically independent of textual information describing goods, thus allowing for filtering non-textual goods (e.g., movies, music).

As a result, niche products that increase the utility of consumers gain a significant share of consumers' demand and become more popular. This phenomenon is known as the Long Tail phenomenon, a term coined by Anderson in [Anderson, 2006]. The long tail of products consists of less popular products that are, typically, not provided by B&M retailers due to opportunity costs and the issue of information overload and overchoice. The distribution of popularity of downloaded song titles at Rhapsody[4] and Wal-mart is depicted in Figure 1.1.

Referring to Figure 1.1, the revenue share of song titles in the long tail is significant, being 40% of total revenue in the case of Rhapsody [Anderson, 2006]. In the case of Amazon.com and Netflix[5], the revenue share of niche products is 25% at Amazon.com for books respectively 21% at Netflix [Anderson, 2006].

[4] `http://www.rhapsody.com`
[5] `http://www.netflix.com`

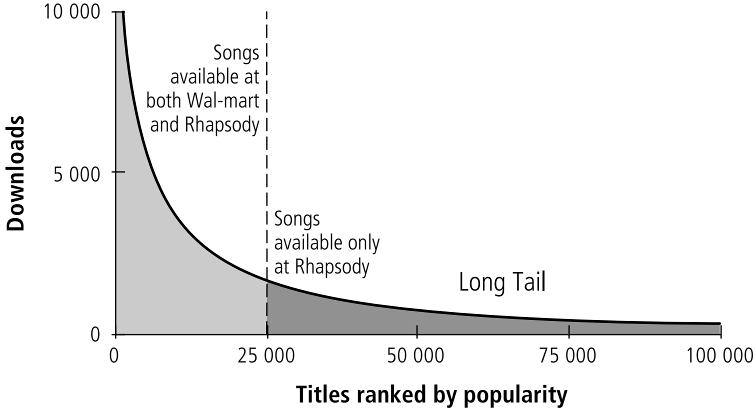


Figure 1.1: Anatomy of the Long Tail at Rhapsody (December 2005) [Anderson, 2006].

Recommender systems are a crucial driver for the Long Tail phenomenon [Hinz et al., 2011] by reducing information overload and overchoice [Anderson, 2006]. As indicated by Brynjolfsson et al. in [Brynjolfsson et al., 2003], the consumer welfare enhanced by $731 million to $1.03 billion in the year 2000 due to the increased product variety of online bookstores. Consumer welfare refers to consumers' total satisfaction. This enhancement in consumer welfare is 7 to 10 times larger than the consumer welfare gain from increased competition and lower prices in the book market [Brynjolfsson and Smith, 2000].

In conclusion, recommender systems are essential for increasing consumer welfare, which ultimately leads to an increase in economic and social welfare.

1.1 Motivation and Thesis

The vast product variety and product variation offered by online retailers provide an amazing amount of choice options to individuals, thus posing a big challenge to them finding and choosing interesting products that provide them the most utility. Recommender systems have emerged to help individuals finding these products.

Recommender systems employ collaborative filtering technology [Burke, 2002, Adomavicius and Tuzhilin, 2005, Ziegler, 2005], a filter technology that considers the product ratings of like-minded individuals to provide recommendations to individuals. In this context, product ratings imply the utility gained by individuals. The fundamental assumption of collaborative filtering is that individuals who previously shared similar preferences behave similarly in the future. Consequently, each individual benefits from the product experience of like-minded individuals.

Online retailers use individuals' product ratings to create preference profiles for the purpose of retrieving like-minded individuals and personalizing content. For this purpose, online retailers collect explicit ratings (e.g., rating score) or implicit ratings (e.g., product browsing history, shopping cart, purchase history). Based on this information, like-minded individuals are commonly identified by comparing the ratings for common rated products [Resnick et al., 1994, Herlocker et al., 1999]. The ratings for common rated products of two individuals comprises both individuals' ratings for all products that both individuals have rated. The basic assumption is that like-minded individuals can be assessed based on the ratings for common rated products.

However, retrieving like-minded individuals based on ratings for common rated products may be inappropriate because ratings for common rated products may not necessarily be a representative sample of two individuals' preferences due to the following four issues:

Significance of similarity. Two individuals' ratings for common rated products may be too scarce to draw a significant conclusion about the preference similarity of both individuals, thus missing like-minded individuals or even mistaking individuals as being like-minded.

Partial representation of preferences. Two individuals' ratings for common rated products correspond to the intersection of both individuals' rated products and therefore may represent only partially both individuals' preferences, which are multifaceted by nature. Consequently, overall preference similarity is, in fact, deduced from partial preference similarity. For instance, both individuals may like the same action movies and rate these in the same way. But, one of both is also interested in romantic movies whereas the other individual is not. As a result, the former rates various romantic movies whilst the latter rates only a few, if at all, romantic movies. Consequently, few or even no romantic movies are element of the set of common rated products, which causes a selection bias. Although both individuals share similar preferences for action movies, a romantic movie may be a good recommendation only for one of both.

Assessability of similarity. The preference similarity between two individuals is not assessable in the case of both individuals not sharing ratings for the same products. Consequently, like-minded individuals are missed due to lack of ratings, which ultimately limits prediction coverage [Herlocker et al., 1999]. Prediction coverage is the percentage of products for which the recommender system is able to generate a recommendation [Ge et al., 2010]. The reason is that collaborative recommendations are limited to products that have been rated by retrieved like-minded individuals.

Incompleteness of preferences. Retailers collect only a fraction of individuals' ratings on their store because individuals purchase products from different stores, above all the non-loyal individuals. Therefore, individuals' ratings are distributed across multiple retailers [Ziegler, 2005, Bannwart et al., 2009], which limits the amount of collected ratings per retailer. Hence, distributed ratings aggravate the issues previously described.

Generally, recommender systems face the problem of having sparse information about individuals' preferences, which leads to poor recommendations. This problem is known as the *cold-start problem*, which commonly affects collaborative filtering-based recommender systems [Schein et al., 2002, Middleton et al., 2002]. As a result of poor recommendations, individuals change to other recommender systems instead of providing further information about their preferences, thus starting the vicious circle of recommender systems keeping to provide poor recommendations.

Machine learning algorithms, in contrast, can be used to efficiently generalize from an individual's ratings to the individual's preferences. These algorithms are able to improve the task of predicting the utility a particular product provides to the individual based on the observations of the individual's product ratings. These algorithms hypothesize individuals' preferences and represent these hypothesized preferences as preference models, which can be used to predict the utility that individuals receive from the consumption of a particular product.

Although these preference models filter relevant products, they lack the ability to find products that provide the most utility to individuals. For instance, an individual may like action movies in general. The individual's preference model filters primarily action movies, which are relevant according to the observed individual's preferences. Nonetheless, the individual does receive different utility from different action movies due to tacit differences between these movies. Furthermore, the individual may occasionally receive high utility from non-action movies, which are not considered as relevant by the individual's preference model.

In conclusion, collaborative filtering is an effective method to provide individuals the products that best fit their preferences. The rating similarity of individuals for common rated products, however, is not an efficient method of retrieving like-minded individuals, especially in the cold-start situation. Hypothesized preferences, in contrast, represent effectively individuals' preferences given that individuals provide sufficient ratings. Hypothesized preferences, however, lack in taking the tacit quality of products into account.

Hence, we claim that using hypothesized preferences of individuals to retrieve like-minded individuals instead of the ratings for common rated products overcomes both the discussed limitations of ratings for common rated products and the limitations of hypothesized preferences. We constitute our claim on the basis of the following four reasons:

- Comparing two hypothesized preferences is independent of the set of common rated products, thus resolving the issue of scarce common rated products (significance of similarity).

- Hypothesized preferences approximate the individual's entire preferences, thus resolving the issue of the set of common rated products representing only partially both individuals' preferences (partial representation of preferences).

- Retrieving like-minded individuals is based on their respective hypothesized preferences, thus resolving the issue of individuals not sharing a single rating for the same product (assessability of similarity).

- Hypothesized preferences can be compared across multiple retailers, thus resolving the issue of distributed ratings and mitigating the cold-start problem (incompleteness of preferences).

In summary, we state our thesis as follows:

Thesis:
Retrieving like-minded individuals based on the similarity of their respective hypothesized preferences (the cause of their product ratings) instead of the similarity of product ratings for common rated products (the effect of their preferences) is effective and efficient, and mitigates the cold-start problem of collaborative filtering.

1.2 Hypothesis-Based Collaborative Filtering in a Nutshell

In this dissertation, we propose *hypothesis-based collaborative filtering* (HCF) to expose individuals to products that best fit their preferences. In HCF, we retrieve like-minded individuals based on the similarity of their respective hypothesized preferences. For this purpose, we use machine learning to hypothesize individuals' preferences. Machine learning is a method to automatically learn to recognize complex patterns from observations [Mitchel, 1997], thus being adequate to hypothesize individuals' preferences from their product ratings.

Generally, the similarity of two individuals' hypothesized preferences can be computed in two different ways. One way is to compare the hypothesized utilities that products provide to both individuals. The hypothesized utility a product provides to a particular individual can be predicted based on the individual's hypothesized preferences. We assume that the more similar both individuals' hypothesized utilities for some products are, the more similar are their respective hypothesized preferences and therefore their preferences. We call this kind of two individuals' preference similarity *hypothesized utility-based preference similarity* (HU preference similarity).

The other way is to compare the composition of both individuals' hypothesized preferences. We assume that the more similar the components of both individuals' hypothesized preferences are, the more similar are their respective hypothesized preferences and therefore their preferences. We call this kind of two individuals' preference similarity *hypothesis composition-based preference similarity* (HC preference similarity).

In this dissertation, we provide the theoretical foundation, the algorithmic framework, and its implementation for both kinds of preference similarity. In both cases, we use machine learning to hypothesize individuals' preferences. For the first kind (i.e., HU preference similarity), we propose to compute the preference similarity of two individuals based on the utilities that some products provide to each of both. For this purpose, we use both individuals' hypothesized preferences to predict the respective utilities of some products. To compute the preference similarity, we propose three similarity metrics to compare product utilities.

For the HC preference similarity, we propose to compute the preference similarity of two individuals based on the compositions of their respective hypothesized preferences. For this purpose, we introduce the notion of *hypothesized partial preference* (HPP), which are self-contained and form the components of hypothesized preferences. An HPP, for example, may specify a high utility gain from recent action movies in which a particular actress acts the heroine. Hence, an HPP is composed of a composition of requirements for product properties and the expected utility gain. To compare the composition of hypothesized preferences, we cross-compare HPPs that constitute the respective hypothesized preferences. Then, we consolidate the similarities for each pair of HPPs to an overall similarity of both hypothesized preferences. We propose several methods to compare HPPs and to consolidate these similarities.

We conduct a large empirical study on a quasi benchmark dataset and diverse variations of this dataset, which vary with respect to sparsity degree, to evaluate the cold-start behavior of HCF. Based on this empirical study, we provide empirical evidence for the superiority of HCF regarding recommendation performance and its robustness against data sparsity relative to state-of-the-art filtering methods.

We use the research methodology of grounded theory for scrutinizing the empirical evaluation to explain the cold-start behavior of HCF regarding the retrieval of like-minded individuals relative to other filtering methods. We formulate a theory about the cold-start behavior of HCF, which we ground on the empirical evaluation, and verify its validity.

1.3 Thesis Statement

To verify our thesis presented in Section 1.1, we rely on the fulfillment of research goals. Prior to discussing the research goals in Section 1.3.2, we present the underlying research hypotheses of these research goals in Section 1.3.1.

1.3.1 Research Hypotheses

In this section, we present the research hypotheses underlying the research goals of this dissertation.

Information gain from domain ontology. A domain ontology is the specification of the conceptualization of domain knowledge [Gruber, 1993]. It formally represents domain knowledge as a set of concepts and their interrelationships. We assume that the efficiency of machine learning can be increased by taking domain ontologies into account when generalizing from observations. We argue that machine learning can enrich observations with domain knowledge to gain additional information to generalize from observations, particularly in the case of insufficient observations. On the other hand, machine learning can exclude some correct generalizations from observations which contradicts to the domain knowledge.

HYPOTHESIS (H1): *The efficiency of machine learning can be increased by taking domain ontologies to gain useful information.*

We verify this hypothesis with respect to an extension of a commonly used machine learning algorithm. We show by means of a synthetic dataset that the efficiency of machine learning can be increased taking a domain ontology, particularly when observations are insufficient.

Ontology-based preference representation. Machine learning algorithms provide different representations of hypothesized preferences, which depends on the machine learning method and its implementation, thus being neither machine readable nor interpretable. We assume that an ontology can be developed, which can formally represent individuals' hypothesized preferences based on a set of appropriate concepts and their interrelationships. We argue that using OWL as the knowledge representation language, individuals' hypothesized preferences are machine readable and semantically interpretable by different recommender systems, which are able to consume semantic information.

HYPOTHESIS (H2): *An ontology is appropriate to specify individuals' hypothesized preferences which is machine readable and therefore can be used as exchange format among different recommender systems.*

We verify this hypothesis with respect to the Web Ontology Language (OWL), which is a knowledge representation language and which provides formal semantics. We show by means of two different machine learning algorithms that hypothesized preferences can be represented with an ontology. For this purpose, we developed an appropriate ontology to specify hypothesized preferences.

Hypothesized utility-based preference similarity. The utility a product provides to individuals depends on their preferences and indicates how well the product fits individuals' preferences. Under the assumption that machine learning algorithms adequately hypothesize individuals preferences, we assume that the more similar utilities for some products of individuals are, the more similar are their preferences. Therefore, predicting the utilities for some products and computing the similarities of utilities for some products allows for the retrieval of like-minded individuals.

HYPOTHESIS (H3.1): *The similarity of hypothesized utilities some products provide to individuals allows for the retrieval of like-minded individuals.*

We verify this hypothesis with respect to a quasi benchmark dataset, which is commonly used to evaluate recommender systems. We provide empirical evidence that *hypothesized utility-based preference similarity* (HU preference similarity) allows for retrieving like-minded individuals.

Hypothesis composition-based preference similarity. The composition of hypothesized preferences in terms of *hypothesized partial preferences* (HPP) represents the cause for hypothesized utilities which products provide to individuals. We assume that by comparing HPPs of different hypothesized preferences we can deduce the similarity of individuals' preferences. Therefore, computing the similarities of the composition of hypothesized preferences allows for the retrieval of like-minded individuals.

HYPOTHESIS (H3.2): *The similarity of the composition of hypothesized preferences in terms of HPPs allows for the retrieval of like-minded individuals.*

We verify this hypothesis with respect to a quasi benchmark dataset, which is commonly used to evaluate recommender systems. We provide empirical evidence that *hypothesis composition-based preference similarity* (HC preference similarity) allows for retrieving like-minded individuals.

Hypothesized partial preference similarity. Individuals may have only partially similar preferences. For instance, two individuals may have similar preferences for Italian food whereas both individuals have different preferences for Asian food. Therefore, both individuals should be considered as like-minded in the case of Italian food, but not in the case of Asian food. We assume that taking *hypothesized partial preference* (HPP) for similar products is more appropriate than considering the overall preference similarity of individuals.

HYPOTHESIS (H3.3): *The partial similarity of hypothesized preferences allows for the case-based retrieval of like-minded individuals.*

We verify this hypothesis with respect to a quasi benchmark dataset, which is commonly used to evaluate recommender systems. We provide empirical evidence that HPP similarity can be more appropriate than overall preference similarity.

Cold-start mitigation. Retrieving like-minded individuals based on the similarity of their ratings for common rated products has some limitations, particularly when individuals provide few ratings. We assume that hypothesizing individuals' preferences by means of machine learning captures more accurately individual preferences, thus still allowing for the accurate retrieval of like-minded individuals. Therefore, *hypothesis-based collaborative filtering* (HCF) should mitigate the cold-start problem, which commonly affects collaborative filtering-based recommender systems.

HYPOTHESIS (H3.4): *Hypothesis-based collaborative filtering mitigates the cold-start problem.*

We verify this hypothesis with respect to a quasi benchmark dataset. We use this dataset to derive some datasets with different severity of the cold-start problem by gradually removing individuals' ratings. We provide empirical evidence that retrieving like-minded individuals based on the similarity of their hypothesized preferences instead of the similarity of ratings for common rated products outperforms the latter method in different cold-start scenarios and even in the case of the quasi benchmark dataset.

Rating predicate. The rating activity of individuals and the received rating activity of products indicates the severity of cold-start problem. We assume that the more severe the cold-start problem is, the more efficient the retrieval of like-minded individuals based on the similarity of their hypothesized preferences is relative to the similarity of their ratings for common rated products.

HYPOTHESIS (H4): *The rating predicate of individuals by means of provided rating activity and products by means of received rating activity indicates the effectivity and the efficiency of retrieving like-minded individuals based on the similarity of hypothesized preferences.*

We verify this hypothesis with respect to the results of our empirical study. We provide empirical evidence that rating predicates of individuals and products can be used to determine the relative efficiency of retrieving like-minded individuals based on the similarities of hypothesized preferences relative the similarities of ratings for common rated products.

1.3.2 Research Goals

We define five research goals as follows:

Domain Ontology-Based Machine Learning Algorithm

RESEARCH GOAL (G1): *Create a machine learning algorithm which takes a domain ontology to increase the efficiency of hypothesizing an individual's preferences.*

Machine learning algorithms generalize from the product ratings provided by an individual to the individual's preferences and builds a preference model, which represents an approximation of the individual's preferences. A machine learning algorithm should be developed which takes a domain ontology into account when hypothesizing the individual's preferences compensate for insufficient product ratings, thus increasing the efficiency of hypothesizing an individual's preferences For the purpose of generality, the domain ontology should be semantically specified with the Web Ontology Language (OWL).

The success of this research goal depends on the acceptance of Hypothesis H1 (information gain from domain ontology hypothesis).

Specification of the Conceptualization of Hypothesized Preferences

RESEARCH GOAL (G2): *Create an preference ontology to semantically describe hypothesized preferences and to mediate an individual's preferences across recommender systems.*

The ontology should specify the concepts and properties to describe an individual's preferences. For this purpose, the ontology should be appropriate for formalizing hypothesized preferences and machine learning models, respectively. The specification of individuals' preferences should be machine readable and semantically interpretable, thus being an appropriate exchange format to mediate individuals' preferences across recommender systems. The Web Ontology Language OWL provides a formal framework for semantically specifying the conceptualization of preferences.

The success of this research goal depends on the acceptance of Hypothesis H2 (ontology-based preference representation hypothesis).

Algorithmic Framework for Comparing Hypothesized Preferences

RESEARCH GOAL (G3): *Develop an algorithmic framework and its implementation to compute the similarity of two individuals' hypothesized preferences.*

An algorithmic framework should be developed to compute the similarity of two individuals' hypothesized preferences. This algorithmic framework should base on a theoretical foundation of comparing hypothesized preferences. The proposed algorithmic framework should be implemented for the purpose of evaluation. An algorithmic framework should be developed and implemented for hypothesized utility-based preference similarity and hypothesis composition-based preference similarity.

The success of this research goal depends on the acceptance of either Hypotheses H3.1 (hypothesized utility-based preference similarity hypothesis), H3.2 (hypothesis composition-based preference similarity hypothesis), or H3.3 (hypothesized partial preference similarity hypothesis).

Effectivity and Efficiency of Hypothesis-Based Collaborative Filtering

RESEARCH GOAL (G4): *Evaluate the cold-start behavior of the effectivity and efficiency of retrieving like-minded individuals based on the similarity of their hypothesized preferences.*

The cold-start behavior of the effectivity and efficiency of retrieving like-minded individuals based on their hypothesized preferences should be evaluated with an empirical study. The cold-start behavior should be analyzed by evaluating the recommendation performance for cold-start situations of different degree. The effectivity and efficiency of hypothesis-based collaborative filtering should be compared to state-of-the-art collaborative filtering approaches.

The success of this research goal depends on the acceptance of either Hypotheses H3.1 (hypothesized utility-based preference similarity hypothesis), H3.2 (hypothesis composition-based preference similarity hypothesis), or H3.3 (hypothesized partial preference similarity hypothesis), and the acceptance of the Hypothesis H3.4 (cold-start mitigation hypothesis).

Grounded Theory of Hypothesis-Based Collaborative Filtering

RESEARCH GOAL (G5): *Develop a theory which explains the characteristics of hypothesis-based collaborative filtering and its cold-start behavior. Based on its characteristics, build a decision model to decide when to favor hypothesized preference similarity over similarity of ratings for common rated products to retrieve like-minded individuals.*

A theory should be developed to explain the characteristics of retrieving like-minded individuals based on their hypothesized preferences. The theory should explain the differences to the retrieval of like-minded individuals given the similarities of their ratings for common rated products. Based on this theory, a decision model should be built to decide when to favor similarities of hypothesized preferences over similarities of ratings for common rated products. The theory and the decision model should be validated by incorporating it in the recommendation performance evaluation. The research methodology of grounded theory should be applied.

The success of this research goal depends on the acceptance of Hypothesis H4 (rating predicate hypothesis).

1.4 Contributions

The conceptual contributions of this dissertation focus on the theoretical foundation and the algorithmic framework to compare hypothesized preferences by means of comparing hypothesized utilities, comparing HPPs and the composition of hypothesized preferences. The technical contributions of this dissertation concentrate on the development of ontologies, tools to retrieve like-minded individuals (i.e., RECOMIZER) and conducting the empirical study and the analysis of its results. The empirical contributions of this dissertation constitutes of two contributions. The first contribution is the empirical evidence of *hypothesis-based collaborative filtering* HCF outperforming other collaborative filtering methods, in cold-start situations in particular. The second contribution is the theory of the efficiency of retrieving like-minded individuals, which is grounded in the empirical results, based on the similarity of their respective hypothesized preferences.

The main contributions of this dissertation are:

1. The theoretical foundation of partial, semi-partial and total similarity of individuals' hypothesized preferences.

2. The extraction of *hypothesized partial preferences* (HPPs) from preference models created by machine learning algorithms and the preference ontology YOULIKE to specify individuals' hypothesized preferences.

3. SEMTREE, an extension of a decision tree induction algorithm which takes a domain ontology to increase the learning efficiency.

4. Two algorithmic frameworks to compute the partial, semi-partial and total similarity of hypothesized preferences. The first algorithmic framework computes the similarity of hypothesized preferences by means of the similarity of their respective hypothesized utilities for some products. The second algorithmic framework computes the similarity of hypothesized preferences by means of the composition of hypothesized preferences in terms of HPPs.

5. An semantic extension of the Jaccard similarity coefficient to compare the semantic similarity sets of semantic concepts.

6. The movie ontology MO, which provides amongst other things a taxonomy of genres.

7. A theory which explains the cold-start behavior of hypothesis-based collaborative filtering and a decision model to support the choice of the most appropriate method of retrieving like-minded individuals.

8. RECOMIZER, a Java implementation of several variations of the two proposed algorithmic frameworks.

9. OMORE, a Firefox extension which allows individuals to incorporate hypothesized utility on Web pages about products across multiple retailers.

1.5 Organization

This dissertation is organized as follows:

Chapter 2 (p.27) reviews related work in the field of recommender systems in general, collaborative filtering, and the cold-start problem in particular.

Chapter 3 (p.41) presents the theoretical foundation to define the similarity of individuals' preferences by means of their respective hypothesized preferences. More precisely, it presents the theoretical foundation of using hypothesized preferences to approximate individuals' preferences and, to this end, introduces the notion of partial preferences. Furthermore, it presents two extraction strategies to extract *hypothesized partial preferences* (HPPs) from decision tree models and probabilistic models. Finally, the preference ontology YOULIKE is presented, which provides the necessary concepts and relations to specify individuals' preferences. This can be used to mediate individuals' preferences across different recommender systems incorporated on different stores.

Chapter 4 (p.53) presents a semantic extension to a common machine learning algorithm, more precisely a decision tree learner, to boost its efficiency of hypothesizing individuals' preferences based on few observations. The accuracy of hypothesized preferences impacts the accuracy of the similarity computation of hypothesized preferences, thus being crucial, especial in cold-start situations in which individuals commonly provide few ratings. More precisely, a domain ontology is used to to provide additional information to boost the efficiency of inducing decision trees. It is shown that the semantic extension for a decision tree learner can approximate more accurately individuals' preferences than comparable decision tree learners.

Chapter 5 (p.65) provides the theoretical foundation of the similarity of hypothesized preferences, the similarity of HPPs and the similarity of *hypothesized semi-partial preferences* (HSPP). Furthermore, it presents two different algorithmic frameworks to compute the similarity of hypothesized preferences based on the similarity of predicted utilities for some products respectively the similarity of hypothesis compositions.

Chapter 6 (p.87) presents the empirical study which we have conducted. This chapter describes the experimental setting, the used dataset, presents and discusses the evaluation results of different implementations of *hypothesis-based collaborative filtering* (HCF) and other candidates for comparison regarding different performance metrics.

Chapter 7 (p.129) presents the analysis of the cold-start behavior of HCF. The research methodology grounded theory to scrutinize the empirical results and to develop a theory to explain the cold-start behavior. Finally, the theory is validated by means of an empirical study.

Chapter 8 (p.189) discusses the conceptual and technical limitations and future work of this dissertation, HCF in particular.

Chapter 9 (p.195) draws the conclusions, discusses the acceptance of the thesis and closes with opportunities for further research.

Appendix A (p.207) summarizes tools we developed for this dissertation. Amongst other things, it presents RECOMIZER, the Java implementation of both algorithmic frameworks.

Appendix B (p.215) presents the movie ontology MO, the taxonomy of movie genres in particular.

Appendix C (p.217) presents the different datasets which we used to conduct the empirical study and to build a theory to explain the cold-start behavior of HCF.

Appendix D (p.225) extends the presented distribution of the recommendation performance of HCF in Chapter 7 by five additional recommender systems.

Appendix E (p.253) presents additional recommendation performance results of HCF and individual properties and product properties for different datasets with different sparsity degree.

Appendix F (p.259) presents additional comparisons of the recommendation performance and cold-start behavior of HCF methods and other collaborative filtering methods.

Appendix G (p.279) presents the foundation of this thesis by means of published work.

2

Related Work

RECOMMENDER SYSTEMS are information systems which provide consumers recommendations regarding products which best fits consumers' preferences, thus reducing information overflow and mitigating overchoice. In fact, recommender systems implement a distinct search paradigm relative to the canonical search paradigm of search engines that is: interesting products find the consumer instead of the consumer search these.

Recommender systems use the collective wisdom of consumers for exposing consumers to products which best fits their preferences. This filtering technology is commonly referred to as collaborative filtering, a term coined by Goldberg [Goldberg et al., 1992].

In the following, we present the formal framework of recommender systems in Section 2.1. Prior to review related work addressing the cold-start problem in Section 2.2.2, we present different approaches of collaborative filtering in Section 2.2. Subsequently, we discuss machine learning in Section 2.3.

2.1 Recommender Systems

2.1.1 Formal Framework

The basic elements for a collaborative filtering based recommender system is the set of individuals I, the set of products G, and the set of ratings R in which individuals explicitly or implicitly state about products. The rating set R is represented as the $m \times n$ rating matrix as it is shown in Figure 2.1 with $m = |I|$ number of individuals and $n = |G|$ number of products.

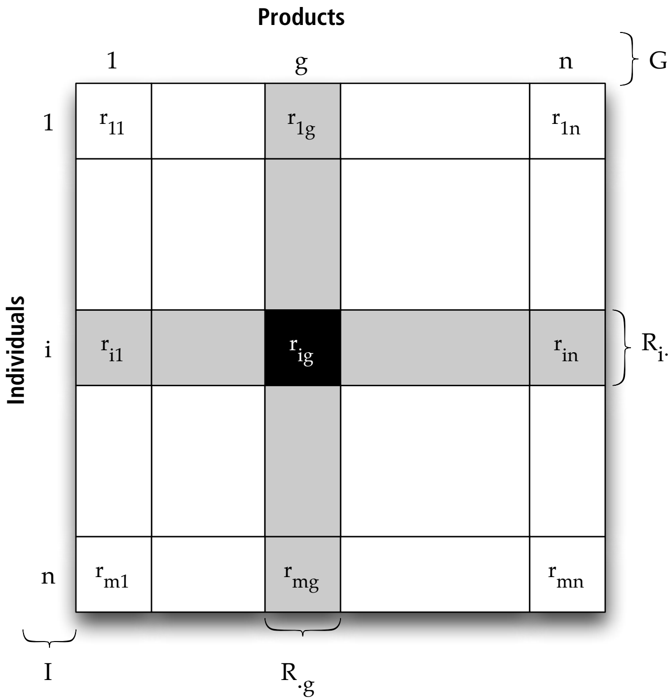


Figure 2.1: Representation of individuals' ratings for all products as the rating matrix R.

We refer to a particular individual as $i \in I$, a particular product as $g \in G$, and a particular rating of individual i for product g as r_{ig}. Note, r_{ig} corresponds to the element at the ith row and gth column of the rating matrix R.

We call the subset of products which individual i has rated with $r \neq \emptyset$ the individual's i rated product set $G_i \subseteq G$. We call the subset of individuals which have rated product g the product's g rated individual set $I_g \subseteq I$. We denote the individual for which we compute the recommendations as the active individual $a \in I$.

Based on the introduced notation, we define a recommender system as a function f which returns for the active individual a the product rating vector $\widehat{R}_{a\cdot}$. The product rating vector $\widehat{R}_{a\cdot}$ provides the individual a's rating r_{ag} for product $g \in G_a$ or a predicted rating $\widehat{r}_{ag}$ for product $g \in G \setminus G_a$. For the computation, the function f considers the active individual a as well as all other individuals in I, all products in G, individual a's product rating vector $R_{a\cdot}$ as well as the rating matrix R and the set of rating concepts C. Hence, we conceptualize a recommender system as follows:

$$\widehat{R}_{a\cdot} = f(a, U, G, R_{a\cdot}, R, C) \tag{2.1}$$

with $\widehat{R}_{a\cdot}$ as the predicted rating vector for the active individual a.

2.1.2 Ratings

We denote the value space of the rating r_{ig} as K which consists of rating concepts C or $\emptyset$ in case of no rating. In other words, $K = C \cup \emptyset$. A particular rating concept is referred to as $c \in C = \{1, \ldots, k-1\}$ with $k = |K|$ number of rating concepts. Generally, a value space is classified to one of the following four groups:

- *Nominal rating*: The task of product recommendation can be treated as a classification problem that associates a product with one ore more rating classes. Popular classes of rating concepts C are {*relevant*, *irrelevant*} or {*likes*, *likes not*}.

- *Ordinal rating*: The rating concepts are interrelated and can be ordered. The typical example for ordinal rating concepts is the star-rating on a 1-5 integer scale: $\{\bigstar, \ldots, \bigstar\bigstar\bigstar\bigstar\bigstar\}$. With ordinal ratings only the assertion can be done that a 4-star rated product is better than a 2-star rated product, but not two times better.
- *Interval rating*: Products can be rated with a numeric value from $\mathbb{R}$. Ratios in this scale are not meaningful. In general, such ratings can be normalized to a $[-1, 1]$ scale.
- *Ratio rating*: Products can be rated with a numeric value from $\mathbb{R}$. In general, such ratings can be normalized to a $[0, 1]$ scale.

2.2 Collaborative Filtering

Collaborative filtering is a method to provide personalized recommendations to an individual by considering the preferences from many other individuals [Resnick et al., 1997, Kautz et al., 1997, Konstan, 2004]. The underlying assumption of collaborative filtering is that individuals who shared the same preferences in the past tend to share the same preferences in the future. To provide personalized recommendations, preferences need to be formalized and conceptualized. Built on this, the similarity of individuals' preferences is computed to retrieve like-minded individuals. Afterwards, collaborative filtering is applied to provide personalized recommendations to a individual by considering prior ratings of like-minded individuals.

Generally, any vector similarity metric can be applied to compute the similarity between product rating vectors. Past research focused mainly on cosine similarity, Pearson correlation, and Spearman's rank correlation [Herlocker et al., 1999, Breese et al., 1998].

In addition to individual preference similarity, [Anand et al., 2007] propose to describe products with semantical concepts and incorporate semantic product similarity to the computation of individual preference similarity. Following the same idea, a multilayer ontology-based hybrid recommendation model is proposed in [Cantador et al., 2008]. Similar products are clustered whereby each cluster represents a topic of interest. Users are related to these clusters to form *communities of interests* [Cantador and Castells, 2006]. Then, all products liked by a community of interests is recommended to its members. However, the computational complexity of nearest neighbor-based collaborative filtering is $O(n^2)$ because the preference similarity of each individual pair has to be computed.

In [Basu et al., 1998], social aspects of the individuals (e.g., age) and content-based information are used to compare individuals' preferences. More precisely, single individual and product features are used to define homogeneous groups of individuals and products. In [Basu et al., 1998], these features are referred to as hybrid features. Basically, partial preference similarity is defined in [Basu et al., 1998] with respect to a single product property. The limitation of this approach is that it limits the partiality to one single, pre-defined product-property.

To address the computational complexity, cluster-based approaches are proposed to reduce the number of comparisons in [Zhang et al., 2008]. Individuals are clustered according to their geographical location. Within these clusters, individuals' preferences are compared. Similarly, clustering individual into groups of similar individuals is suggest in [Rashid et al., 2006] with ClustKNN. Then, similarity is computed only between the centroids of these clusters. It is suggest to use k-Means clustering algorithms to control the tradeoff of computational complexity and recommendation accuracy which depends on the number of clusters. Empirical evidence is given in [Rashid et al., 2006] that clustering techniques do not sacrifice much recommendation accuracy but making collaborative filtering scalable.

Given its reliance on product vector ratings, however, ClustKNN does not taking in-depth preference similarity into account.

Predicting missing ratings to improve matrix factorization approaches is proposed in [Melville et al., 2002]. Nonetheless, the preference similarity is biased by the product set. For instance, when a product set is dominated by one product type, then individuals' preferences are mainly compared with respect to this single product type.

The Fab system [Balabanovic and Shoham, 1997] recommends documents. To this goal, the individual generates his individual profile by specifying his topics of interests. Those individual profiles are used to compute individual's with similar topic of interests. The Fab system then recommends the individual documents that match the individual's profile and that have been liked by similar individuals. A more sophisticated way of defining individual profiles is proposed in [Good et al., 1999]. User profiles are expressed by preference vectors that contain the relevance of certain topics. Documents are filtered according to the individual profiles. *Collaborative filtering* based on individual similarity is used to filter the retrieved documents. The relevance of product features is used in [Anand et al., 2007] as preference vectors instead.

In [Middleton et al., 2002] the synergy between ontologies and recommender systems has been demonstrated. Quickstep [Middleton et al., 2004] uses an ontology for individual profiling. It learns by observing the individual's behavior in what research domain the individual is interested in and recommends further papers of that research domain. The individual profile consists of a preference vector containing the relevance of the corresponding category in the ontology.

Referral Web [Kautz et al., 1997] is solely based on social networks. It guides the individual through the social network to an expert that may help. Trust networks on top of social networks have been proposed as good sources for *collaborative filtering*[Golbeck et al., 2003]. The positive correlation between trust and preferences has been investigated in [Ziegler and Lausen, 2004].

Ziegler proposed taxonomic filtering, taxonomy diversification and trust networks to overcome the challenges of decentralized recommender systems. In [Berkovsky et al., 2007], a different approach of a decentralized recommender system is proposed. They propose the mediation of individuals' preference models across different domains to enhance personalization of individual models. It is empirically shown, that like-minded individuals in one a domain are likely to be like-minded in another domain. For instance, individuals sharing similar preferences for music are likely to share similar preferences for movies. The disadvantage of these approaches is that they are limited to product rating vectors and cannot handle partial preference similarity.

Multiple ontologies are used to build a decision tree in [Zhan et al., 2002]. The training procedure uses a top-down approach. In a first step, the top most concept of the ontologies are compared according to their information gain. The concept with the greatest information gain is used as decision node. In the next steps, the information gain of all the sub-concepts of the concept used in the decision tree are calculated and compared with the remaining concepts of the other ontologies. This procedure is repeated until every concept of all ontologies are used. In the context of assigning GO terms to proteins, only one ontology is used.

In [Kurzynski, 1983], Kurzynski introduces a decision tree learner that combines features to decrease the probability of misclassification. The optimal combination of features is formalized as an optimization problem, where all feature combinations are computed. That is problematic because of runtime complexity as discussed in [Safavian and Landgrebe, 1991].

Domain ontologies are used to extract features to describe products in [Kudenko, 2000]. The extracted features are then used in conjunction with machine learning algorithms. In our approach we already have the features.

In addition, we generalize features during the training procedure to improve the result.

2.2.1 General Framework for Collaborative Filtering

[Herlocker et al., 1999, Resnick et al., 1994] propose a general framework for a nearest neighbor-based collaborative filtering approach, which is presented in Eq. (2.2). The general framework specifies a method to predict the unobserved rating $\widehat{r}_{ig}$ of individual i for product g considering the k most like-minded individuals. The goal is to predict a rating $\widehat{r}_{ig}$ for individual i and product g based on his/her average rating $\overline{r}_i$ and of k nearest neighbors' (i.e., other k individuals with most similar preferences) weighted rating deviation $r_{ng} - \overline{r}_n$ of that product.

$$\widehat{r}_{ig} = \overline{r}_i + \frac{\sum\limits_{n=1}^{k} w_{in} * (r_{ng} - \overline{r}_n)}{\sum\limits_{n=1}^{k} w_{ig}} \tag{2.2}$$

Whilst many similarity metrics for the computation of $w_{a,b}$ have been proposed, [Herlocker et al., 1999] show that the Pearson correlation outperforms the Spearman's rank correlation and the cosine similarity.

2.2.2 Cold-Start Problem

Recommender systems commonly face the cold-start problem when recommendations are required for individuals or products for which too little information (i.e., ratings) is known. Given that individuals only use a recommender system that provides reasonable recommendations, providing poor recommendations will lead to attrition of individuals. Hence, a recommender system may never achieve a critical mass of ratings to provide reasonable recommendations. Thus, the challenge is to provide reasonable recommendations even with little information about the individuals' preferences.

According to [Middleton et al., 2002], three different types of the cold-start problems can be distinguished:

new-system cold-start. The new-system cold-start refers to the initial stage of a recommender system when no or only few initial ratings are provided. Every recommender system performs poorly in this situation because it lacks crucial information to build accurate preference profiles of individuals. Even content-based recommender systems need a few observations of the individuals' interest to provide reasonable recommendations.

new-user cold-start. The new-user cold-start refers to the situation when too few ratings for an individual exist even in the light of sufficient information about others. Note that a similar problem may arise when ratings for some products are extremely sparse as the resulting set of common rated products may be small. A collaborative filtering-based recommender system faces the same problem with individuals providing ratings for products which have rarely been rated by other individuals.

new-item cold-start. The new-item cold-start refers to the problem that arises with products that have no or only few ratings yet. The products provide only a small set of potential like-minded individuals.

In the following, we denote a situation with very sparse information a major and a situation with more information a minor cold-start problem situation.

Generally, content-based approaches are incorporated to face the cold-start problem. Content filtering approaches can be combined with collaborative filtering approaches to build hybrid recommender systems. A classification and survey of hybrid recommender systems is found in [Burke, 2002]. One of the first hybrid recommender systems is the Fab system [Balabanovic and Shoham, 1997], which recommends documents. Its individuals are asked to create an individual profile by selecting topics of interest. Users are similar if they share many topics of interest. Documents are recommended when they match the individual's profile and have been liked by individuals with similar individual profiles. Whilst this approach addresses the cold-start problem its results are too generic and lack precision.

Another approach is to preprocess the data before replacing missing ratings with predicted ones. [Melville et al., 2002], for instance, predict unknown individual ratings based on known ratings. For that purpose, an individual's preferences is learned (or hypothesized) with a machine learning algorithm that predicts the ratings for not yet rated products.

The mediation of individual models across different domains is proposed in [Berkovsky et al., 2007] for enhanced personalization of individual models. In fact, domains are defined as product features in [Berkovsky et al., 2007]. They empirically show that like-minded individuals in one domain are likely to be like-minded in another domain. For instance, individuals sharing similar preferences for music are likely to share similar preferences for movies. The disadvantage of these approaches is that they are limited to product rating vectors and cannot handle partial preference similarity. In other words, individuals are considered as generally like-minded if they share similar preferences for one domain. However, this approach lacks accounting for individuals having only partially similar preferences.

Generally, individuals differ in their rating behavior such that some individuals rate products on average higher than other individuals. Hence, it is shown in [Bell and Koren, 2007] that normalizing the individuals ratings improves the accuracy of computed individual preference similarities and consequently results in more accurate recommendations.

2.3 Machine Learning

Machine learning is a branch of artificial intelligence. It is concerned with algorithms which enable computers to learn from observations. A computer program is said to learn from past experience E if its performance in some tasks T is improved with more experience E [Mitchel, 1997]. According to [Mitchel, 1997], the learning problem of learning the individual's preferences needs to be well-defined. For that purpose, the following three features need to be defined: the class of tasks, the measure of performance to be improved and the source of experience:

- Task T: For the individual i, predict for the product $g \in G$ the proper rating concept $c \in C$ such that $c = r_{ig}^{true}$
- Performance measure P: e.g., Percentage of correct predictions of rating concepts $c = r_{ig}^{true}$ of products $g \in G_i$ for the individual i
- Experience E: The individual i's product rating vector $R_{i.}$ and the set of products G_i

The learner faces the problem of hypothesizing the rating concepts $c \in C$ for the product $g \in G_i$. Hence, the learner has to find the hypothesis $h^i(g)$ from the hypothesis space H of all possible hypotheses which predicts best the rating concept c the individual i associates with product g. For this purpose, the performance measure P is used to determine the best hypothesis h^i. This hypothesis $h^i(g)$ is the hypothesized rating function and the individual's hypothesized preferences respectively. A perfect hypothesized rating function $h^i(g)$ for $g \in G_i$ is defined as:

$$h^i(g) = u^i(g) \quad , \forall g \in G_i \tag{2.3}$$

The performance of $h^i(g)$ depends on both the hypothesis space H, which is based on human designer's choice, and the amount of Experience E and the number of the individual i's rating.

II

Preference Modeling

3

Conceptualization and Specification of Preferences

THE individuals' preferences are multifaceted by nature; in other words individuals prefer different products for different reasons. Therefore, we need to formalize these preferences to support the retrieval of like-minded individuals, which ultimately improves personalized recommendations based on collaborative filtering.

We introduce the notion of partial preferences to take the multiplicity of facets of individual preferences into account. Based on partial preferences, we can allow for fine-grained comparison of two individuals.

In the following, we develop the theoretical foundation for modeling the individual's preferences as a set of partial preferences in Section 3.1. Subsequently, we explain how to extract partial preferences from machine learning models in Section 3.2 followed by the presentation of the preference ontology YOULIKE to specify hypothesized preferences in Section 3.3. Finally, we discuss the acceptance of Hypothesis H2 in Section 3.4 (ontology-based preference representation hypothesis).

Part of this chapter have been published in [Bouza et al., 2009] and [Bouza and Bernstein, 2012]

3.1 Formalization of Preferences

We formalize the individual's preferences in terms of elements, which are typically found in recommender systems. These elements are the set of individuals I, the set of products G an the set of ratings R. The ratings are expressed either explicitly or implicitly by individuals. The set of rating concepts C defines the value space which can be realized by ratings. At this point, we refer to Section 2.1.2 for possible sets of rating concepts C. We refer to a particular individual as $i \in I$, to a particular product as $g \in G$, to a particular rating concept as $c \in C$ and we denote the rating of an individual i for product g as $r_{ig} \in R$.

We assume that every individual i is able to rate every product g in terms of rating concepts C. Based on this assumption, we define individual i's preferences as a personal utility function $u^i : G \rightarrow C$, which maps a given item g to a rating concept c:

$$u^i : g \mapsto c \equiv r_{ig} \quad , \forall g \in G \tag{3.1}$$

whereby the realization of a particular rating concept c for $u^i(g)$ is the rating r_{ig}. Note that the personal utility function u^i is not necessarily surjective or injective. The personal utility function u^i is a total function for all individuals, however, which maps every $g \in G$ to exactly one $c \in C$.

A computer program is able to learn the individual i's utility function $u^i(g)$ based on individual i's ratings. According to [Mitchel, 1997], a computer program is said to learn from ratings if the accuracy of estimating the correct rating improves. The goal of a computer program can be defined as finding the hypothesis $h : G \rightarrow C$ from a hypothesis space H such that $h(g) \approx u^i(g)$ for all products $g \in G$. The hypothesis space H is based on human designer's choice.

Typically, a computer program finds a less adequate hypothesis h when the hypothesis space H is missing the correct hypothesis or individual i

expresses too few ratings. We denote this issue with the error term $\epsilon^i(g)$. Therefore, the relation between $u^i(g)$ and $h^i(g)$ is defined as:

$$u^i(g) = h^i(g) + \epsilon^i(g) \tag{3.2}$$

However, given an adequate hypothesis space H, we argue that the error term $\epsilon^i(g)$ goes to 0 for more ratings, because a *learning* computer program selects by definition a more accurate hypothesis h with more experience E [Mitchel, 1997], i.e., when individual i expresses more ratings. Hence, we formulate the following corollary:

Corollary 1. *The individual i's preferences or rather individual i's utility function $u^i(g)$ can be approximated with the appropriate hypothesized utility function $h^i(g)$:*

$$u^i(g) \approx h^i(g) \tag{3.3}$$

3.1.1 Partial Preferences

An individual's preferences are multifaceted by nature such that an individual likes products with different properties. We incorporate this aspect into the individual preference model by introducing the notion of partial preferences. According to Eq. (3.1), we model individual i's preferences as a utility function with case distinctions, whereby each case or rather condition D^i_j with $j \in \{1, \ldots, z\}$ and z as the number of cases corresponds to the partial preference $u^i_j(g)$:

$$u^i(g) = \begin{cases} u^i_1(g) = c_v & \text{if } g \text{ satisfies condition } D^i_1 \\ & \vdots \\ u^i_z(g) = c_w & \text{if } g \text{ satisfies condition } D^i_z \end{cases} \tag{3.4}$$

with $c_v, c_w \in C$. Note that every partial preference $u^i_j(g)$ is a partial function $u^i_j : G \rightsquigarrow C$ and is defined only for $g \in G_{D^i_j} \subseteq G$, the subset of G which satisfies condition D^i_j.

From Corollary 1 and Eq. (3.4), we deduce that $h^i(g)$ is likewise a function with case distinctions, whereby each case corresponds to a hypothesized partial preference (HPP) $h^i_j(g)$:

$$h^i(g) = \begin{cases} h^i_1(g) = c_v & \text{if } g \text{ satisfies condition } D^i_1 \\ & \vdots \\ h^i_s(g) = c_w & \text{if } g \text{ satisfies condition } D^i_s \end{cases} \tag{3.5}$$

with the partial preference $u^i_j(g)$ having $h^i_j(g)$ as correspondent. Hence, we formulate the following corollary:

Corollary 2. *The individual i's partial preference or rather individual i's partial utility function $u^i_j(g)$ can be approximated with the appropriate hypothesized partial utility function $h^i_j(g)$:*

$$u^i_j(g) \approx h^i_j(g) \tag{3.6}$$

We assume an individual's preferences to be consistent. Under this assumption, an individual rates every product which satisfies the same condition the same. Analogously, the hypothesized individual preferences estimates the same rating for products which satisfy the same condition.

3.2 Partial Preference Extraction from Machine Learning Models

Referring to Eq. (3.5), the product's properties determine which condition is satisfied, thus determining the proper HPP. We interpret a condition as a conjunction of constraints on the product's properties and the rating as the conclusion. We represent a conjunction of constraints as a set of constraints whereby each constraint has to be fulfilled. We refer to a set of constraints with p.

Typically, different machine learning algorithms represent a hypothesis differently, thus encoding HPPs differently. In the following, we discuss how these HPPs are encoded in decision tree models and Naïve Bayes probabilistic models and how to extract these from both types of models.

3.2.1 Partial Preference Extraction from Decision Tree Classifier

A decision tree learner represents a hypothesis as a decision tree in which each node corresponds to a test of some property of a product. Each edge corresponds to a possible evaluation of such a test. A test in combination with an evaluation specifies a constraint. Thus, each branch which starts at the root node and ends at a leaf corresponds to a conjunction of constraints with the leaf as the conclusion. Therefore, an HPP is encoded as a branch in a decision tree which starts at the root and ends in some leaf.

Figure 3.1 presents an exemplified decision tree representation of an individual's preferences. The highlighted path in Figure 3.1 corresponds to an HPP and consists of the evaluated tests $\{P, A, T\}$ and its conclusion respectively utility h.

To extract all HPPs, we have to parse all possible branches. In total, we get as many HPPs as the number of leafs of the decision tree.

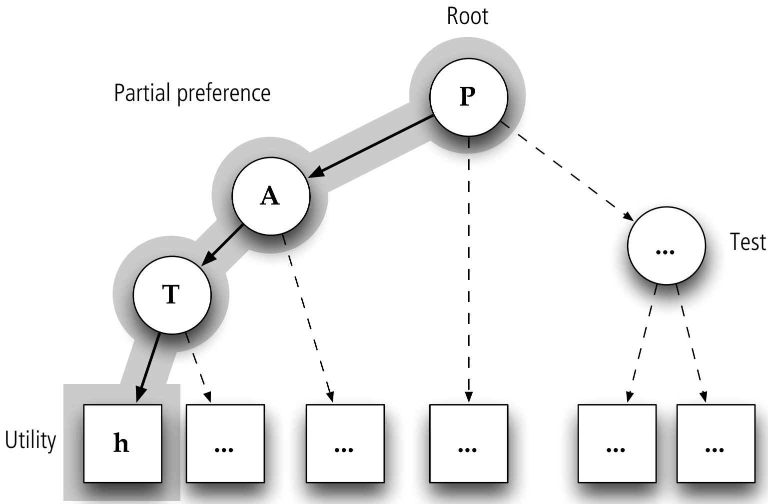


Figure 3.1: Partial preferences encoded as branches the root to the leaf of the decision tree. The nodes of a branch corresponds to tests of some product properties and the leaf corresponds to the utility of the product.

3.2.2 Partial Preference Extraction from Naïve Bayesian Classifier

A Naïve Bayes classifier is a probabilistic classifier based on applying Bayes's theorem. It models a hypothesis as a conditional probability model $P(C|A_1, \ldots, A_n)$ with C as the rating concept variable and A_j as the jth property variable of the set of all properties A. To predict a product's utility or rather rating, we can use the Naïve Bayes classification rule:

$$C \leftarrow \arg\max_{c_k} P(C = c_k) \prod_j P(A_j|C = c_k) \tag{3.7}$$

This rule calculates the most probable utility based on the observed probability distribution of $P(C)$ and $P(A_j|C)$.

The Naïve Bayes classification rule is, in fact, a linear combination of all conditional probabilities of $P(C)$ and $P(A_j|C)$. Since Naïve Bayes assumes strong (i.e., naïve) independence among properties, an individual's preferences for product properties are likewise independent. Therefore, we can interpret each conditional probability $P(C) * P(A_j|C)$ as an HPP with the set of constraints consisting of the single constraint A_j as the condition and the most probable conclusion c as the utility:

$$C \leftarrow \arg\max_{c_k} P(C = c_k) * P(A_j|C = c_k) \tag{3.8}$$

For the purpose of simplification, we apply Bayes's theorem to simplify the definition of HPPs of Eq. (3.8) and rewrite HPPs as:

$$C \leftarrow \arg\max_{c_k} P(C = c_k|A_j) \tag{3.9}$$

To extract all HPPs, we compute for each attribute the most probable utility.

3.3 Ontological Specification of Hypothesized Preferences

We developed the YouLike ontology[1] to provide a controlled vocabulary to semantically formalize an individual's hypothesized preferences. We specify the YouLike ontology with the Web Ontology Language OWL, whose main purpose is to capture the semantics of information and make it comprehensible to different machines. For this purpose, we followed the ontology development guide suggested by [Noy and McGuiness, 2001].

[1] http://www.preferenceontology.org

Specifying the YOULIKE ontology with OWL makes it appropriate to mediate an individual's preferences among different recommender systems and Semantic Web agents. We define three concepts and five properties to describe hypothesized preferences. These concepts and properties are presented in Table 3.1 and Table 3.2, respectively. In the following, we describe the reason behind and the usage of the elements of the YOULIKE ontology.

Concepts	**Description**
`Profile`	corresponds to the individual's preferences.
`Preference`	corresponds to an individual's partial preference.
`Constraint`	specifies a constraint on the product's properties.

Table 3.1: Concepts of the YOULIKE ontology, which specify the elements of the conceptualization of preferences.

Properties	**Description**
`features`	an individual features a `Profile`
`considers`	a `Profile` considers a `Preference`
`rated`	the utility of a `Preference` is rated with a numeric value.
`requires`	a `Preference` requires a `Constraint`.
`excludes`	a `Preference` excludes a `Constraint`

Table 3.2: Properties of the YOULIKE ontology, which are used to specify an individual's preferences based on the concepts of the YouLike ontology.

An individual's preferences varies depending on the situation and the context. For instance, an individual may be limited in capital, thus optimizing his utility with cheaper products. The YOULIKE ontology accounts for this by allowing an individual for featuring multiple preference profiles. The YOULIKE ontology provides the concept `Profile` to model multiple preference profiles. Figure 3.2 shows an example of an individual's hypothesized preferences being formalized with the YOULIKE ontology. As

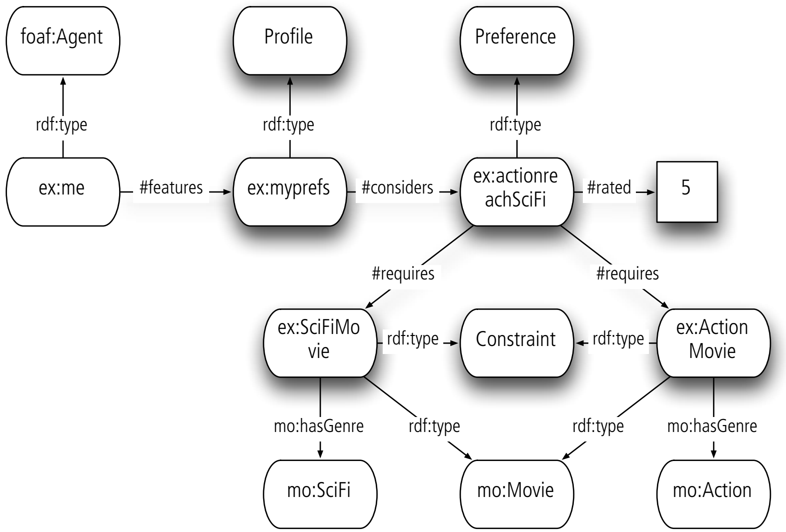


Figure 3.2: Illustration of an exemplified formalization of an individual's hypothesized preferences using the YOULIKE ontology.

shown in Figure 3.2, the individual `ex:me` features the preference profile `ex:myprefs`. The YOULIKE ontology provides the property `features` to relate a preference profile to an individual. We use the individual's FoaF profile, which is commonly used in the Semantic Web.

An HPP is formalized as an instance of the YOULIKE class `Preference` and is, in fact, independent of a preference profile. Therefore, it can be used by multiple profiles of multiple individuals. As illustrated in Figure 3.2, the preference profile `ex:myprefs` considers the HPP `ex:actionreachSciFi`. The YOULIKE ontology provides the property `considers` to relate an HPP to a preference profile.

As described in Section 3.2, an HPP consists of a set of constraints and a conclusion. Constraints are formalized as an instance of the YOULIKE class

`Constraint`. As shown in Figure 3.2, the HPP `ex:actionreachSciFi` requires two constraints. As demonstrated in Figure 3.2, constraints can be defined with other ontologies. We use the movie ontology MO[2] (see Appendix B) to specify the required properties of a movie. The YOULIKE ontology provides the properties `requires` and `excludes` to specify required properties respectively excluded properties of a product.

Products which fulfill the constraints of some HPPs provide a utility to an individual. We formalize this utility as a numerical value. The YOULIKE ontology provides the property `rated` to evaluate the utility of an HPP. As illustrated in Figure 3.2, the HPP `ex:actionreachSciFi` is rated with the numeric value 5.

3.4 Acceptance of Hypotheses

We presented the theoretical foundation of using machine learning to hypothesize individuals' preferences. We introduced the notion of partial preferences and presented the preference ontology YOULIKE. In the following, we discuss the acceptance of the following stated hypothesis:

Ontology-based preference representation hypothesis (H2). Based on the theoretical foundation and the notion of partial preferences, we demonstrated how to extract HPPs from two different representations of hypotheses, namely decision tree model and Naïve Bayes model. Based on the preference ontology YOULIKE, we demonstrated how to specify hypothesized preferences.

Since we can specify hypothesized preferences with the preference ontology YOULIKE, we conclude that it is feasible to specify hypothesized preferences with ontologies, the preference ontology YOULIKE in particular.

Hence, we accept H2.

[2] `http://www.movieontology.org`

3.5 Summary

In this chapter, we have developed the theoretical foundation for modeling the individual's preferences as a set of partial preferences and explained how to extract partial preferences from machine learning models. Before using this theoretical foundation to define the similarity of hypothesized preferences, we introduce in the next chapter an extension to a machine learning algorithm to enable accurate hypothesized preferences even if individuals provide few ratings.

4

Domain Ontology-Boosted Decision Tree Induction

MACHINE LEARNING can be used to efficiently generalize from an individual's ratings to the individual's preferences. Machine learning algorithms are able to learn to improve the task of predicting the utility of a particular product for the individual from the observations of the individual's product ratings by finding a appropriate hypothesis. These algorithms represent the learned knowledge or rather hypothesized preference about the individual's preferences as a preference model.

However, the accuracy of these preference models or rather hypothesized preferences depends on the amount of learning data (i.e., ratings). Especially in the case of individuals providing few ratings, machine learning may not be appropriate to accurately hypothesize individuals' preferences.

Background knowledge is a valuable source to compensate for insufficient ratings individuals may provide [Mitchel, 1997]. In the following, we present the decision tree induction method SEMTREE, which uses domain knowledge in form of a domain ontology to compensate for insufficient

Part of this chapter has been published in [Bouza et al., 2008] product feedback and to boost the learning. The automated integration of concept features from the ontology increases dynamically the amount of features used in the built decision tree by integrating single concept features on purpose. Therefore, no pre-computation of concept features is needed that leads to a biased training of decision trees. In addition, no selection of significant concept features for the feature vector representation has to be done. Thus, no domain expert is needed to select the significant concept features manually and no additional computation with, for example, single-stage Bayes rule [Safavian and Landgrebe, 1991] is needed to extract them automated beforehand.

Furthermore, SEMTREE dynamically adjusts the values of the considered concept features. Ultimately, SEMTREE does not increase the feature vector representation. Even though SEMTREE considers all concept features, the use of concept features is purposeful and well directed.

In the following, we describe the basics of decision tree induction in Section 4.1. Subsequently, we describe in depth the mechanisms of using background knowledge in the training process. Finally, we discuss the acceptance of Hypothesis H1 (Information gain of domain ontology hypothesis). in Section 4.3.

4.1 Decision Tree Induction

A decision tree is a simple recursive data structure to express a sequential classification process [Quinlan, 1986]. The basic idea of a decision trees is to sort items down to certain leafs according to the values in the feature vector that describes the item. The leaves of the decision tree then provide the associated classification of the item. It can be understood as applying a sophisticated strategy of accessing the significant features to associate the item with the proper classification. Many strategies exists to build decision trees and are summarized in [Safavian and Landgrebe, 1991].

The common way to learn decision trees is to train them with a training set which consists of feature vectors and their classification. The decision tree is then applied to a test set to evaluate, prune and optimize the decision tree. Decision trees are built by using features as decision nodes.

4.1.1 Feature Selection

The basic idea of building decision trees is to select the features that splits the set of instances into subsets containing as much similar instances as possible in the sense of common classification. The selection of the most significant features and the splitting of instances results in ordered subsets which contain instances with mainly a common classification. Decision trees like ID3, C4.5 and C5.0 [Quinlan, 1993] respectively are successful examples of this approach. They basically apply information theory to select the features that reduces the entropy.

Let's denote $A \in \mathcal{A}$ as feature with $\mathcal{A}$ as the set of all features. Further we denote $\mathcal{V}_A$ as the set of all possible values of feature A and $v \in \mathcal{V}_A$ as the value of the feature A. We define $\mathcal{S}$ as the set of all instances and $\mathcal{S}_v$ as the set of instances with the characteristic v, i.e. the set of all instances where attribute A has value v. The probability p_i describes the distribution of all instances to the corresponding classification by looking at the ith value v in $\mathcal{V}_A$ of the attribute A.

The entropy of a set $\mathcal{S}$ of instances is calculated with Eq. (4.1).

$$Entropy(\mathcal{S}) = \sum_{i=1}^{n} -p_i * log_2 p_i \tag{4.1}$$

The difference of the entropy of $\mathcal{S}$ and the summed and weighted entropy of the sets $\mathcal{S}_v$ of the feature A is calculated as in Eq. (4.2).

$$Gain(\mathcal{S}, A) = Entropy(\mathcal{S}) - \sum_{v \in \mathcal{V}_A} \frac{\mathcal{S}_v}{\mathcal{S}} * Entropy(\mathcal{S}_v) \tag{4.2}$$

If the entropy of feature A is high or low then $Gain(\mathcal{S}, A)$ is high or low, respectively.

A high information gain a of feature A is interpreted as split of the instances according to their classification. Finally, a decision tree is build by selecting the feature with the highest information gain and append it as a decision node to the decision tree. The instances are then spitted into subsets where the same procedure is applied until no split is possible or no feature reduces the entropy anymore. The last decision node is then declared as the leaf and contains the distribution of instances to their associated class.

4.2 SEMTREE Extension to the Decision Tree Model

In SEMTREE we map features from the feature vector to concepts in the ontology (classes or ontology instances). This way SEMTREE profits from the class membership and class hierarchy information and therefore uses this background information for the computation of recommendations. For the background knowledge SEMTREE uses RDFS/OWL ontologies [Dean and Schreiber, 2004]. Compared to original decision trees, SEMTREE takes advantage form the explicit semantics of the ontology (i.e., the relationship between the ontology concepts) to improve the built decision tree and increase the accuracy of item classification.

4.2.1 Basic Idea

The basic idea of SEMTREE is, that we look which feature provides the greater information gain: The feature itself, or the super concept in the ontology which the feature is mapped to. The super concepts form the ontology that summarizes its sub-concepts (classes or instances which represent features) are called *concept features*. More details on concept features are discussed in the description of the SEMTREE algorithm below.

4.2.2 Injecting Concept Features to Generalize from Features

The logic of feature generalization by injecting concept features is presented in Figure 4.1. The concepts that can be used for the concept feature generation are given by an ontology. We use the *rdfs:subClassOf* and *rdf:type* properties which connect instances with concepts and concepts among each other concepts.

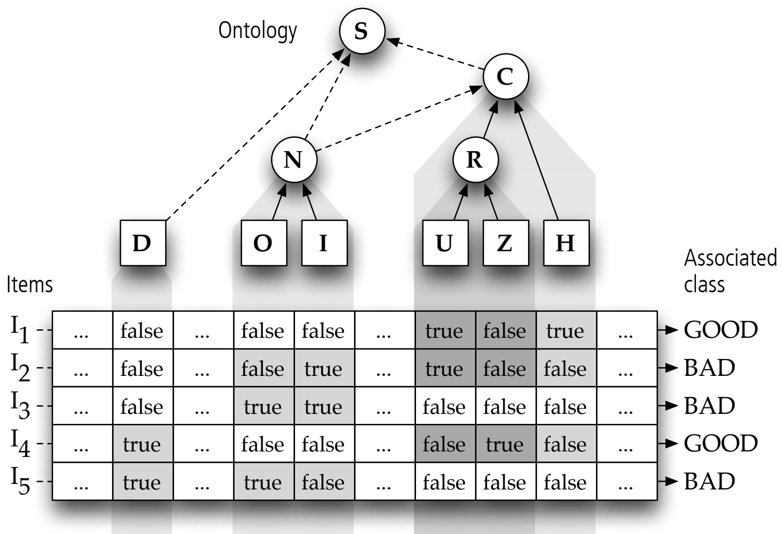


Figure 4.1: SEMTREE generalizing semantically from single concepts using an ontology to improve the classification.

Let's denote $A_i \in A_1, \cdots, A_n$ as the feature vector representation of item

I_i. The associated classification to item I_i is denoted as $C_i \in C_1, \cdots, C_n$. Further, $\{D, O, I, U, Z, H\}$ represent ontology instances whereas $\{S, C, R, N\}$ classes and superclasses.

If we look at the two instances where the feature value for feature D is set to `true` (I_4 and I_5) the instance is once classified to class $GOOD$ and the other time to BAD. Therefore, the feature D does not provide evidence for the classification of the instance.

The features O and I individually are similar to the case before and do not provide evidence for the instance classification. Since O and I are instances of the ontology class N we can combine (generalize) both features to the concept feature of N, which is used as decision node in the tree only if it provides a greater information gain. To calculate the information gain, we have to set the values of the concept feature for each instance. If one of the feature values for an instance is set to `true` the feature value of N is also set to `true`. This is depicted in the figure with the grey background. In boxes with grey background color, the feature value is set to `true`, otherwise `false`.

In the next case we go a step further. Again, the features U, Z, and H do not provide evidence for the classification. However, if we look at the concept features R and C the information gain is not greater than the one of the individual features and we discard the concept features and use the features U, Z, and H instead. In the example in Figure 4.1 the feature or rather concept feature with the highest information gain is the concept feature of UN and is therefore used as decision node in the tree.

4.2.3 Classification

The classification by the SEMTREE classifier is similar to the common classification procedure in decision trees. SEMTREE sorts the classifying item according to the associated feature vector representation down to the right leaf. The leaf contains a classification distribution and provides the classification with the highest probability. A classifier sorts items down to the appropriate leaf and decides according to associated feature at every decision node which path to the lower subtree should be taken. Since the SEMTREE classifiers contain injected concept features, the decision which path to the lower subtree is different. To calculate the value for a concept feature value, we follow down the ontology hierarchy to the individual features. If at least one of the feature values is set to `true` the corresponding concept feature value is set to `true`, otherwise to `false`. This corresponds to and or connection of all involved feature values. If the value is determined, the selection of the proper lower subtree is taken according to the common procedure of decision trees.

4.2.4 Implementation

The Algorithm 4.1 shows the recursive algorithm for inducing the SEMTREE classifier. The SEMTREE classifier is built by selecting the feature with the highest information gain and use it as a decision tree node. The instances are split according to this node. For each of these set of instances, the feature with the highest information gain is used again and added as child node. This process is repeated until instances cannot be split again or the information gain of features is zero.

Next, the feature or feature-superclass with the highest information gain is used as decision tree node and to split the item instances into the two sets. For both sets of item instances the algorithm continues to select the feature or superclass with the highest information gain to build a new subtree recursively until no splitting of the item instances are possible.

Algorithm 4.2 shows, how to incorporate the domain ontology to retrieve features, which are candidates of a decision tree node. In a first step, the algorithm calculates for every feature its information gain by splitting the (item) instances into two sets. The first set contains all instances that provide the feature and the second set contains all item instances that do not.

Algorithm 4.1 Build SEMTREE classifier

Require: $|instances| > 0$

```
procedure BUILDCLASSIFIER(instances)
    node ← FINDBESTSPLIT(instances)
    if node = null then
        return null                          ▷ no split, end of branch
    end if
    instsplit ← splitInstances(node, instances)
    for all instset in instsplit do
        if |instset| > 0 then
            node.add(BUILDCLASSIFIER(instset))
        end if
    end for
    return node
end procedure
```

In a second step, the algorithm retrieves a list of all related concepts or rather superclasses for every feature of the domain ontology. For each related concepts, SEMTREE calculates its information gain. In depth, the algorithm splits the item instances into two sets again. The first set contains item instances that provide at least one feature that is an instance of the superclass. The second set contains all item instances that do not provide any feature that are instances of the superclass.

4.3 Acceptance of Hypotheses

We presented the decision tree induction extension SEMTREE in Section 4.2 and discussed by means of an example in Section 4.2.2 the information gain of semantic concepts. In the following, we discuss the acceptance of the following stated hypotheses.

Algorithm 4.2 Find best splitting feature

Require: $|instances| > 0$
Require: $features, ontology$

```
procedure FindBestSplit(instances)
    maxGain ← 0                                   ▷ highest information gain
    maxFeature ← null             ▷ feature with highest information gain
    for all feature in features do
        candidate ← feature
        gain ← computeInformationGain(feature)
        for all relFeature in ontology.getRelated(feature) do
            semGain ← computeInformationGain(relFeature)
            if semGain > gain then
                gain ← semGain
                candidate ← relFeature
            end if
        end for
        if gain > maxGain then
            maxGain ← gain
            maxFeature ← candidate
        end if
    end for
    if maxGain > 0 then
        return createNode(maxFeature)
    else
        return null
    end if
end procedure
```

Hypothesized utility-based preference similarity hypothesis (H1). We demonstrated and discussed by means of an example in Section 4.2.2 how a domain ontology can help in the context of machine learning to improve the efficiency of generalizing from few observations. We showed that, for instance, a taxonomic relationships among feature concepts which describe products provide a better interpretation of individuals' ratings, thus generalizing more efficiently from individuals' ratings to the individuals' preferences.

Since domain ontologies can provide additional information which makes machine learning more efficient, we conclude that, besides of observations, machine learning gains information from domain ontologies.

Therefore, we accept H1.

4.4 Summary

In this chapter, we have introduce a semantic extension to a decision tree learner to enable accurate hypothesized preferences of individuals, which provide few ratings. In the next chapter, we use the theoretical foundation from the previous chapter to define the similarity of hypothesized preferences and develop two algorithmic frameworks to compute the similarity of hypothesized preferences.

III

Preference Similarity

5

Hypothesized Preference Similarity

INDIVIDUALS' preferences are different, thus assessing the personal utility from various products differently. Despite the variety of preferences, individuals are , typically, more similar to some individuals than to others by means of preferences. In this thesis, we refer to individuals with similar preferences as like-minded individuals.

Collaborative filtering considers an individual's most similar individuals in terms of preferences to provide the individual with useful product recommendations. Therefore, the primary task in collaborative filtering is to retrieve like-minded individuals. Typically, the preference similarity between two individuals is computed based on the product feedback for the set of common rated products.

However, the set of common rated products provides a limited examination of both individuals preferences, which we discussed in Section 1.1. To recapitulate, the issues are the significance of sparse common rated products, partial representation of both individuals' preferences, assessability of similarity based on no common rated products and the distribution of feedback across retailers.

Part of this chapter have been published in [Bouza et al., 2009] and [Bouza and Bernstein, 2012]

We propose to retrieve like-minded individuals based on the similarity of their respective preferences, the cause of their product feedbacks, instead of the similarity of product feedbacks for common rated products, which represent partially the effect of their preferences. As discussed in Chapter 3, we use machine learning to generalize individuals' preferences from their product feedback and represent individuals with the corresponding hypothesized preferences or preference model respectively.

Generally, the similarity of hypothesized preferences of two individuals can be computed in two different ways. One way is to evaluate both individuals' utility for several products based on their respective hypothesized preferences. We deduce the preference similarity of both individuals from the predicted utility similarity for the evaluated products. We refer to this method as *black-box similarity calculation*, since we restrict the preference comparison to the relationship of products and their appropriate predicted utility, which corresponds the input-process-output (IPO) model.

The other way is to analyze the composition of both individuals' hypothesized preferences. For this purpose, we introduce the notion of partial preferences to take the multiplicity of facets of individual preferences into account and to compose hypothesized preferences with. We deduce the preference similarity of both individuals from the pairwise comparison of the appropriate *hypothesized partial preferences* (HPPs) of both individuals' hypothesized preferences. In contrast to the black-box similarity calculation, we consider the composition of hypothesized preferences, which consists of HPPs. Therefore, we refer to this method as *white-box similarity calculation*.

In Section 5.1, we present the theoretical foundation of the similarity of two individuals' hypothesized preferences. Furthermore, we provide the theoretical foundation for HPPs and *hypothesized semi-partial preferences* (HSPPs). Subsequently, we present our two proposed approaches (i.e., black-box similarity calculation and the white-box similarity calculation) of how to compare hypothesized preferences. We present the first approach (i.e., black-box similarity calculation) in Section 5.2. This approach computes the similarity of two hypothesized preferences based on their respective predicted utility for several products. We present the second approach (i.e., white-box similarity calculation) in Section 5.3. This approach computes the similarity of two hypothesized preferences based on their respective hypothesis composition, the set of HPPs.

5.1 Theoretical Foundation of Hypothesized Preference Similarity

In this section, we provide the theoretical foundation of hypothesized preference similarity. For this purpose, we define the preference similarity of two individuals $a \in I$ and $b \in I$ as the similarity function $sim : I \times I \rightarrow \mathbb{R}$ which maps the pair of individuals $(a, b) \in I \times I$ to a real value $[0, 1] \subset \mathbb{R}$:

$$sim : (a, b) \in I \times I \rightarrow [0, 1] \subset \mathbb{R} \tag{5.1}$$

A similarity value of 1 refers to individuals who share identical preferences and a similarity value of 0 means no relation between both individuals' preferences.

With respect to Eq. (3.1), we can express the preference similarity between two individuals a and b, $sim(a, b)$, as the similarity of their respective personal utility functions u^a and u^b. Hence,

$$sim(a, b) = sim\big(u^a(g), u^b(g)\big) \tag{5.2}$$

Typically, neither the personal utility function $u^a(g)$ nor $u^b(g)$ are known. However, without loss of generality, we can use Eq. (3.2) and write:

$$sim\big(u^a(g), u^b(g)\big) = sim\big(h^a(g) + \epsilon^a(g), h^b(g) + \epsilon^b(g)\big) \tag{5.3}$$

Due to Corollary 1, we can approximate both utility functions $u^a(g)$ and $u^b(g)$ with the hypothesized rating function $h^a(g)$ and $h^b(g)$, respectively. Consequently, we can approximate the similarity between both personal utility functions by the similarity between both hypothesized utility functions. Hence, we formulate the following corollary:

Corollary 3. *The similarity of individuals a and b preferences or rather both utility functions $u^a(g)$ respectively $u^b(g)$ can be approximated with the similarity of the appropriate hypothesized utility functions $h^a(g)$ respectively $h^b(g)$:*

$$sim\big(u^a(g), u^b(g)\big) \approx sim\big(h^a(g), h^b(g)\big) \tag{5.4}$$

5.1.1 Hypothesized Partial Preference Similarity

Referring to Section 3.1.1, the individuals a and b's preferences are modeled as utility functions $u^a(g)$ respectively $u^b(g)$ with case distinction with each case D_i^a respectively D_j^b corresponding to partial preferences $u_i^a(g)$ respectively $u_j^b(g)$. We denote the set of partial preferences of individuals a and b as V^a respectively V^b. Hence, we define the partial preference similarity of two individuals a and b with respect to their partial preferences $u_i^a \in V^a$ and $u_j^b \in V^b$ as the similarity function $sim : V^a \times V^b \rightarrow \mathbb{R}$ which maps the pair of partial preferences $(u_i^a, u_j^b) \in V^a \times V^b$ to a real value $[0,1] \subset \mathbb{R}$:

$$sim : (u_i^a, u_j^b) \in V^a \times V^b \rightarrow [0,1] \subset \mathbb{R} \tag{5.5}$$

A similarity value of 1 refers to individuals who share identical partial preferences and a similarity value of 0 means no relation between both individuals' partial preferences.

Due to Corollary 2, we can approximate both partial utility functions $u_i^a(g)$ and $u_j^b(g)$ with the hypothesized utility function $h_i^a(g)$ and $h_j^b(g)$, respectively. Consequently, we can approximate the similarity between both personal utility functions by the similarity between both hypothesized utility functions. Hence, we formulate the following corollary:

Corollary 4. *The similarity of individuals a and b partial preferences or rather both partial utility functions $u_i^a(g)$ respectively $u_j^b(g)$ can be approximated with the similarity of the appropriate hypothesized partial utility functions $h_i^a(g)$ respectively $h_j^b(g)$:*

$$sim\big(u_i^a(g), u_j^b(g)\big) \approx sim\big(h_i^a(g), h_j^b(g)\big) \tag{5.6}$$

5.1.2 Hypothesized Semi-Partial Preference Similarity

Referring to Referring to Section 3.1.1, partial preferences are partial functions. This means that these partial preferences are only defined for a subset of products. Due to this fact, predicted utility-based partial preference similarity is not defined according to the definition of HPP similarity in Section 5.1.1. For this purpose we denote

We denote the set of partial preferences of individual a as V^a and the set of all preferences as V . Hence, we define the semi-partial preference similarity of two individuals a and b with respect to individual a's partial preferences $u_i^a \in V^a$ and individual b's preference $u^b \in V$ as the similarity function $sim : V^a \times V \to \mathbb{R}$ which maps the pair of partial preference and preferences $(u_i^a, u^b) \in V^a \times V$ to a real value $[0,1] \subset \mathbb{R}$:

$$sim : (u_i^a, u^b) \in V^a \times V \to [0,1] \subset \mathbb{R} \tag{5.7}$$

Note that for products $g \in G_a$ which satisfy the condition D_i^a the semi-partial preference similarity is a total function. We denote the set of products $g \in G_i$ which individual i rated and satisfy condition D_j^i with $G_{D_j^i}$.

From Corollary 1, Corollary 2, Corollary 3 and Corollary 4, we can deduce that the semi-partial preference similarity is approximated with the HSPP similarity. Hence, we formulate the following corollary:

Corollary 5. *The semi-partial preference similarity of individuals a and b with respect to individual a's partial preference u_i^a and individual b's preference u^b*

can be approximated with the hypothesized semi-partial preference *(HSPP) similarity of the appropriate* hypothesized partial preference *(HPP)* $h_i^a(g)$ *of individual a and the hypothesized preferences* $h^b(g)$ *of individual b:*

$$sim\big(u_i^a(g), u^b(g)\big) \approx sim\big(h_i^a(g), h^b(g)\big) \tag{5.8}$$

5.2 Hypothesized Utility-Based Preference Similarity

In this section, we present the first method of comparing two individuals' preferences which is based on the similarity of the hypothesized utility of both individuals' hypothesized preferences for several products. This method is a black-box similarity calculation method because it considers exclusively the effect of hypothesized preferences in terms of utility products provide.

In the following, we present three different similarity functions to compute the similarity of predicted utilities of two hypothesized preferences, $sim\big(h^a(g), h^b(g)\big)$, which we defined in Eq. (5.4). The first similarity function computes the correlation of predicted utilities of several products. It is presented in Section 5.2.2. The second similarity function computes the probability of identical predicted utilities of several products. It is presented in Section 5.2.3. The third similarity function or rather partial similarity function computes the probability of identical predicted utilities of several products. It is presented in Section 5.2.4.

5.2.1 Product Set for Utility Prediction

Prior to comparing predicted utilities for several products, we need to specify the set of products on which we base the comparison of predicted utilities. In the following, we discuss three candidates for the set of products to consider for the comparison of predicted utilities. We denote the set of products for which individual i provides feedback as $G_i \subseteq G$.

The first candidate is the set of common rated products, which is commonly used in collaborative filtering-based recommender systems. The set of common rated products of two individuals a and b is defined as

$$G_{a \cap b} \equiv G_a \cap G_b \tag{5.9}$$

However, the set of common rated products $G_{a \cap b}$ may not be appropriate to compare the preference similarity of both individuals a and b as we discussed in Section 1.1, specifically in a cold-start situation.

The second candidate is the set of all products G, which provides the most variety of products. However, the huge size of the set G makes it inappropriate due to the computational effort required for predicting and comparing the utilities for all products. Furthermore, retailers provide different assortment, thus requiring additional effort to retrieve all products to predict utilities for.

The third candidate is the set of products for which either of both individuals a and b provides feedback. We refer to this set as the set of unified rated products, which we define as:

$$G_{a \cup b} \equiv G_a \cup G_b \tag{5.10}$$

The set $G_{a \cup b}$ provides more information to conclude preference similarity compared to the set of common rated products $G_{a \cap b}$ due to $G_{a \cap b} \subseteq G_{a \cup b}$. Furthermore, every feedback is considered, thus accounting for both individuals' entirely observed preferences.

Therefore, we propose to use the set of unified rated products $G_{a \cup b}$ for the similarity calculation of both individuals a and b.

5.2.2 Correlative Predicted Utility-Based Similarity

If the feedback type of C is either an interval rating or a ratio rating as defined in Section 2.1.2, then we can exploit the notion of correlation that is defined on both scales. For this purpose, we need to formulate the individual i's utility of products, which is defined by individual i's preferences respectively utility function $u^i(g)$, as a random variable, similar to [Chajewska and Koller, 2000]. Thus, we formulate individual i's utility of products as the random variable U^i:

$$U^i(g) = u^i(g) \tag{5.11}$$

Hence, we define the preference similarity of two individuals a and b as the correlation of both random variables U^a and U^b:

$$sim(a, b) \equiv corr(U^a, U^b) \tag{5.12}$$

Empirical evidence is provided in [Herlocker et al., 1999] that Pearson's correlation is more appropriate than Spearman's rank correlation. Therefore, we use Pearson's correlation to measure the similarity of both individuals' hypothesized preferences. In this context, the Pearson's correlation ρ_{U^a,U^b} is defined as the covariance of both individuals' utility random variable U^a and U^b divided by the product of their corresponding standard deviations σ_{U^a} and σ_{U^b}

$$\rho_{U^a,U^b} = \frac{cov(U^a, U^b)}{\sigma_{U^a}\sigma_{U^b}} \tag{5.13}$$

whereby $cov\big(U^a, U^b\big)$ is defined as the expected value of both random variables U^a and U^b and their respective expected values μ_a and μ_b

$$\begin{aligned} cov\big(U^a, U^b\big) &= E\big[(U^a - \mu_{U^a})(U^b - \mu_{U^b})\big] \\ &= \frac{1}{|G_{a\cup b}|} \sum_{g \in G_{a\cup b}} \big(u^a(g) - \overline{u^a}\big)\big(u^b(g) - \overline{u^b}\big) \end{aligned} \tag{5.14}$$

with $\overline{u^i}$ as the average utility for individual $i \in I$. The standard deviation σ_{U^i} is defined as

$$\begin{aligned}\sigma_{U^i} &= \sqrt{E\left[(U^i - \mu_{U^i})^2\right]} \\ &= \sqrt{\frac{1}{|G_i|} \sum_{g \in G_i} \left(u^i(g) - \overline{u^i}\right)}\end{aligned} \tag{5.15}$$

Hence, we define the similarity of two individuals a and b as the Pearson's correlation between both individuals' preferences:

$$sim(a,b) = \frac{cov\left(U^a, U^b\right)}{\sigma_{U^a}\sigma_{U^b}} \tag{5.16}$$

To compute the similarity defined in Eq. (5.16), we need to hypothesize U^i. According to Corollary 1, the individual i's utility function $u^i(g)$ can be approximated with the appropriate hypothesized utility function $h^i(g)$, thus enabling the similarity computation of both individuals' preferences.

Thus, we define the hypothesized utility random variable H^i analogously to Eq. (5.11) as:

$$H^i(g) = h^i(g) \tag{5.17}$$

Based on Corollary 1, we conclude from Eq. (5.11) and Eq. (5.17) that H^i approximates U^i, likewise. Hence,

$$U^i \approx H^i \tag{5.18}$$

From Eq. (5.16) and Eq. (5.18), we deduce the preference similarity of two individuals a and b as the correlation of both corresponding hypothesized utility random variables H^a and H^b. Hence, we approximate the preference similarity of individuals a and b as follows:

$$sim(a,b) \approx \frac{cov\left(H^a, H^b\right)}{\sigma_{H^a}\sigma_{H^b}} \tag{5.19}$$

5.2.3 Probabilistic Predicted Utility-Based Similarity

If the feedback type of C is a nominal class as defined in Section 2.1.2, then we can treat the evaluation of a product g's utility for an individual i as a classification problem. Thus, we can define the similarity of two individuals a and b as the probability that both individuals' utility of the same products are identical. For this purpose, we need to formulate the individual i's utility of products, $u^i(g)$, as a random variable U^i. At this point we refer to Eq. (5.11) where we defined the utility random variable U^i already.

Hence, we define the preference similarity of two individuals a and b as the probability that both individuals's utility of the same products are identical:

$$sim(a,b) = P(U^a = U^b) \tag{5.20}$$

Referring to Eq. (5.18), we can approximate U^i with H^i:

$$P(U^a = U^b) \approx P(H^a = H^b) \tag{5.21}$$

Hence, we can approximate the probabilistic predicted utility-based similarity of both individuals a and b as follows:

$$sim(a,b) \approx P(H^a = H^b) \tag{5.22}$$

Note that the target domain of this probabilistic similarity function is limited on the interval $[0,1] \subset \mathbb{R}$ with 0, no similar preferences, and 1, identical preferences.

5.2.4 Probabilistic Predicted Utility-Based Semi-Partial Similarity

Likewise to the previous section, we define the semi-partial preference similarity of two individuals a and b with respect to individual a's partial preference u^a_i and individual b's preference u^b as the probability that both individuals' utility of the same products $g \in G_{D_i^a}$ are identical. For this purpose, we need to formulate the individual i's utility of products, $u^i(g)$, as a random variable U^i. At this point we refer to Eq. (5.11) where we defined the utility random variable U^i already. Furthermore, we formulate individual i's utility of products which satisfy the partial preference u_j^i as the random variable U_j^i. Note that the random variable U_j^i is, in fact, a constant, because a product fitting partial preference u_j^i provides in any case the same utility according to the definition of partial preferences in Section 3.1.1. Hence, we define the semi-partial preference similarity of two individuals a and b with respect to individual a's partial preference u_i^a and individual b's preference u^b as the probability that both individuals' utility of the same products $g \in G_{D_i^a}$ are identical:

$$sim\big(u_i^a(g), u^b(g)\big) = P(U_i^a = U^b) \tag{5.23}$$

Referring to Eq. (5.18), we can approximate U^i with H^i:

$$P(U_i^a = U^b) \approx P(H_i^a = H^b) \tag{5.24}$$

Hence, we can approximate the probabilistic predicted utility-based semi-partial similarity of both individuals a and b as follows:

$$sim(u_i^a, u^b) \approx P(H_i^a = H^b) \tag{5.25}$$

5.3 Hypothesis Composition-Based Preference Similarity

In this section, we present the second method of comparing two individuals' preferences which is based on the similarity of the composition of the hypothesized preferences of both individuals. This method is a white-box similarity calculation method, because we consider the inner structure of hypothesized preferences.

As defined in Eq. (3.5), a hypothesized rating function $h^i(g)$ is defined as a function with case distinctions: I. e. $h^i(g)$ corresponds to a set of HPPs. In the following, we represent this set of individual i's *hypothesized partial preferences* (HPPs) as the vector v^i and denote the set of all HPPs as the H. Consequently, we define the similarity of both individual a's and b's preferences as the similarity between both vectors v^a and v^b. For this purpose, we define the function $v^a \circ v^{b\top} : H^m \times H^n \to \mathbb{R}^{m \times n}$ with $m = card(v^a)$ and $n = card(v^b)$ as the number of partial preferences of individual a respectively b. This function maps the pair of sets of HPPs $(v^a, v^b) \subset H$ to $\mathbb{R}^m \times \mathbb{R}^n$, the partial preference similarity matrix S

$$v^a \circ v^{b\top} \mapsto S \tag{5.26}$$

, which is depicted in Figure 5.1.

Each element s_{ij} of the partial preference similarity matrix S corresponds to the similarity of individual a's HPP $h^a_i(g)$ and individual b's HPP $h^b_j(g)$. Therefore:

$$s_{ij} = sim\big(h^a_i(g), h^b_j(g)\big) \tag{5.27}$$

Based on Eq. (5.26), we define the similarity between both hypothesized preferences $h^a(g)$ and $h^b(g)$ as the similarity function $sim(v^a \circ v^{b\top}) : \mathbb{R}^{m \times n} \to [0,1] \subset \mathbb{R}$, which consolidates all partial preference similarities in S. Hence, the theoretical framework of our algorithmic framework is:

$$sim\big(h^a(g), h^b(g)\big) \equiv sim(v^a \circ v^{b\top}) \tag{5.28}$$

Our algorithmic framework of *hypothesis composition-based preference similarity* (HC preference similarity) consists of two components. One component is a method to compute the similarity of HPPs to determine S and

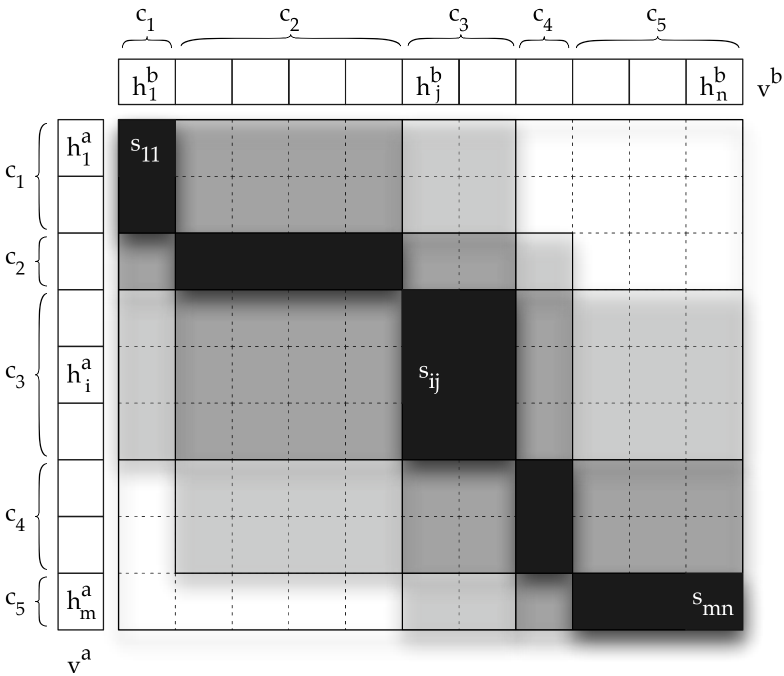


Figure 5.1: The partial preference similarity matrix S representing the similarities between the HPPs of two individuals a and b.

refers to Eq. (5.26). This component is presented in Section 5.3.1.

The other component is a method to consolidate the similarities of S and is refers to Eq. (5.28). This component is presented in Section 5.3.2.

5.3.1 Similarity of Hypothesized Partial Preferences

As defined in Section 3.2, an HPP $h_i(g)$ consists of a set of constraints p_i and the assigned rating c_i. Based on this, we define the similarity between two HPP $h^a_i(g)$ and $h^b_j(g)$ as the similarity between both corresponding constraint set p_i^a and p_j^b combined with the similarity between both corresponding ratings c_i^a and c_j^b:

$$sim\big(h_i^a(g), h_j^b(g)\big) \equiv sim(p_i^a, p_j^b) * sim(c_i^a, c_j^b) \tag{5.29}$$

As it is depicted in Figure 5.1, the similarity $sim(p_i^a, p_j^b) * sim(c_i^a, c_j^b)$ corresponds to the element s_{ij} in the partial preference similarity matrix S, i.e., $s_{ij} = sim(p_i^a, p_j^b) * sim(c_i^a, c_j^b)$.

This algorithmic framework allows for any kind of similarity metrics to compute the similarity between constraint sets and the similarity between rating concepts. In the following, we propose two possible similarity metrics for computing $sim(p_i^a, p_j^b)$ and a similarity metric for computing $sim(c_i^a, c_j^b)$.

Similarity of Constraint Sets

In the following, we propose two similarity metrics to compute the similarity of two constraint sets. Firstly, we propose the Jaccard similarity coefficient, which is commonly used to compare the similarity of two sets. Secondly, we propose an semantic extension of the Jaccard similarity coefficient to take the semantic similarity of constraints in different sets into account.

Jaccard similarity coefficient. The Jaccard similarity coefficient is a commonly used method to compare two sets with respect to their similarity and diversity. This similarity coefficient is applicable since we represent conjunction of constraints as constraint sets (see Section 3.2). By applying the Jaccard similarity coefficient, we implicitly consider two conjunctions of constraints increasingly similar the more constraints they have in common. Hence, we define the similarity of two constraint sets $sim(p_i^a, p_j^b)$ as the Jaccard similarity coefficient $J(p_i^a, p_j^b)$. In our context, the Jaccard similarity coefficient is defined as the size of common constraints divided by the size of the union of all constraints. Thus, the similarity between two set of constraints is:

$$sim(p_i^a, p_j^b) \equiv J(p_i^a, p_j^b) = \frac{|p_i^a \cap p_j^b|}{|p_i^a \cup p_j^b|} \tag{5.30}$$

Semantic extension of Jaccard similarity coefficient. The Jaccard similarity coefficient does not take the semantic similarity of constraints into account. Therefore, the Jaccard similarity coefficient is lacking to consider semantic similarity, thus considering two sets as less similar as these are from a semantic perspective.

For this reason, we extended the definition of the Jaccard similarity coefficient to allow for the comparison of the similarity of sets consisting of semantic concepts. We consider two sets as increasingly similar the more semantically similar the constraints of both sets are. These constraints are based on concepts which are defined by an semantic ontology. We define the semantic extension of the Jaccard similarity coefficient as:

$$J_{sem}(p_i^a, p_j^b) \equiv \frac{|p_i^a \cap_{sem} p_j^b|}{|p_i^a \cup_{sem} p_j^b|} \tag{5.31}$$

We reformulate the size of the union $|p_i^a \cup_{sem} p_j^b|$ as the summed size of both sets $|p_i^a|$ and $|p_j^b|$ minus the size of the semantic intersection $|p_i^a \cap_{sem} p_j^b|$

$$|p_i^a \cup_{sem} p_j^b| = |p_i^a| + |p_j^b| - |p_i^a \cap_{sem} p_j^b| \tag{5.32}$$

and interpret the semantic intersection $|p_i^a \cap_{sem} p_j^b|$ as the semantic similarity of the constraints of both sets. For this purpose, we define the semantic intersection as the sum of the maximal similarity of one constraint of one set with each constraint of the other set and multiply the sum with $\frac{1}{2}$ to account for redundancy:

$$|p_i^a \cap_{sem} p_j^b| \equiv \frac{1}{2}\Big(\sum_{x \in p_i^a} max\big(sim(x, p_j^b)\big) + \sum_{y \in p_j^b} max\big(sim(y, p_i^a)\big)\Big) \tag{5.33}$$

with $sim(z, p_j^i) : P \times P^{card(p_j^i)} \rightarrow \mathbb{R}^{card(p_j^i)}$ with P as the set of all constraints. At this point, we want to emphasize that when no semantic similarity exists but similarity and diversity, then $|p_i^a \cap_{sem} p_j^b| = |p_i^a \cap p_j^b|$ holds.

To compute the semantic similarity $sim(z, p_j^i)$ of constraint z and the constraints in p_j^i, we need to incorporate the notion of semantic similarity of two concepts. An overview of the state-of-the-art in semantic similarity measures of concepts is provided in [Pirró and Euzenat, 2010, Bernstein et al., 2005]. At this point, we want to emphasize that any semantic similarity measure can be used which compute the similarity between two concepts. We present one measure, which is originally proposed in [Wu and Palmer, 1994]. This measure considers the hierarchical relationship of concepts, which is defined in by an ontology. It is defined as follow

$$sim_{Wu\&Palmer}(x, y) = \frac{2 * depth(lcs(x, y))}{depth(x) + depth(y)} \quad , x \in p_i^a, y \in p_j^b \tag{5.34}$$

whereby $lcs(x, y)$ corresponds to the least common subsumer of the concepts x and y

Similarity of Rating Concepts

The similarity between rating concepts depends on the relation among these. In the following, we present one such similarity measure.

Rating equality similarity. In the case that rating concepts are classes or categories respectively, we can define the rating concept similarity $sim(c_i^a, c_j^b)$ as:

$$sim(c_i^a, c_j^b) \equiv \begin{cases} 1 & \text{if } c_i^a = c_j^b \\ 0 & \text{otherwise} \end{cases} \tag{5.35}$$

Note that the black colored fields in Figure 5.1 indicate the cases for which the rating concepts are equal.

This algorithmic framework allows for other similarity measures for rating concepts. We want to emphasize that taking the rating similarity into account can result into a more accurate partial preference similarity matrix S. For instance, two HPPs may be identical or similar but the corresponding ratings are semantically very close. Accounting for such cases can be more appropriate. Figure 5.1 indicates these potential cases as grey colored fields.

5.3.2 Similarity Computation Based on Partial Preference Similarity Matrix

The partial preference similarity matrix S contains all similarity values among HPPs. We define three consolidation steps to consolidate these similarity values. In the first step, we define a function $f(h_i, v) : \mathbb{R}^{card(v)} \rightarrow \mathbb{R}$, which consolidates all similarity values for the given HPP h_i. In fact, we apply this function to every row and column in S. Finally, this results in two vectors w^a and w^b for individuals a and b. The element w_i^a corresponds the consolidated similarity of the HPP h_i^a and b's set of HPPs.

In the second step, we consolidate the two vectors w^a and w^b by computing the arithmetic mean of all values for each vector. The two resulting means describe the similarity between one entire preference model or rather hypothesized preferences and the other one. These two values need to be consolidated because we consider in the algorithmic framework similarity as symmetric.

In the last step, we consolidate both similarity values with the harmonic mean. We use the harmonic mean because it is appropriate to average rates.

To summarize, we define $sim(v^a \circ v^{b\top})$ as:

$$sim(v^a \circ v^{b\top}) \equiv \frac{\frac{2}{card(v^a)}\left[\sum_{h_i^a \in v^a} f(h_i^a, v^b)\right] * \frac{1}{card(v^b)}\left[\sum_{h_j^b \in v^b} f(h_j^b, v^a)\right]}{\frac{1}{card(v^a)}\left[\sum_{h_i^a \in v^a} f(h_i^a, v^b)\right] + \frac{1}{card(v^b)}\left[\sum_{h_j^b \in v^b} f(h_j^b, v^a)\right]} \tag{5.36}$$

In the following, we propose two functions to consolidate the similarity values for an HPP.

Supremum norm. The supremum norm (or infinity norm) is defined over a set of numeric values and it is equals the greatest value of the set. In our context, the supremum norm corresponds to the greatest similarity of an HPP of individual a to any other HPP of individual b. Given the HPP h_i^a of individual a, the set of HPPs v^b of individual b, and the set of similarity values $S_{i\cdot}$ of h_i^a and each HPP $h_j^b \in v^b$, we define $f(h_i^a, v^b)$ as follows:

$$f(h_i^a, v^b) \equiv \left|\left|v^b\right|\right|_\infty = max(||s_{i1}||, \ldots, ||s_{in}||) \quad , n = card(v^b) \tag{5.37}$$

Noisy-Or gate. The noisy-or gate considers every member of a set to be likely the cause of a particular effect [Pearl, 1988]. Therefore, the noisy-or operator is the probability of an effect given the probabilities of the members being the cause. In our context, the similarity values are considered as independent probabilities causing both individuals a and b being similar regarding HPP h_i^a of individual a. Given the HPP h_i^a of individual a, the set of HPPs v^b of individual b, and the set of probabilities $S_{i\cdot}$ of each HPP $h_j^b \in v^b$ and the HPP h_i^a being similar, we define $f(h_i^a, v^b)$ as follows:

$$f(h_i, v^b) = 1 - \prod_{s_{ij} \in S_{i\cdot}} (1 - s_{ij}) \tag{5.38}$$

5.4 Summary

In this chapter, we have presented two algorithmic frameworks to compute the similarity of hypothesized preferences. In the next chapter, we conduct an empirical study to provide empirical evidence of the superiority of both presented algorithmic frameworks.

IV

Evaluation

6

Evaluation

We conducted an empirical study to verify our hypotheses H3.1 (hypothesized-utility-based preference similarity hypothesis), H3.2 (hypothesis composition-based preference similarity hypothesis), H3.3 (hypothesized partial preference similarity hypothesis) and H3.4 (cold-start mitigation hypothesis). For this purpose, we evaluated the performance of our implementation of different variations of our two proposed hypothesis-based methods described in Section 5.2 and Section 5.3 to retrieve like-minded individuals based on the comparison of the similarity of hypothesized preferences.

In the following, we first present the experimental setting in Section 6.1. Subsequently, we present the candidates to be evaluated in Section 6.2 and the datasets to evaluate the cold-start behavior in Section 6.3. Then, we present and discuss the results in Section 6.4. Prior to discussing the acceptance of the hypotheses in Section 6.6, we compare the information gain from product ratings and hypothesized preferences in Section 6.5.

6.1 Experimental Setting

To evaluate our hypotheses from Section 1.3.1, we performed k-fold cross-validation with $k = 5$ for each of the ten datasets. In k-fold cross-validation, the dataset is randomly partitioned into k folds. One fold is held out as the testing set for testing the model and the remaining $k - 1$ folds are used as the training set for building the model. This process is repeated k times with each of the folds held out exactly once as the testing set and the remaining $k - 1$ folds as the training set. Finally, the k testing results are averaged to produce one result. We perform k-fold cross-validation to mitigate any bias caused by choosing samples from the dataset for training and testing which are not representative [Witten and Frank, 2005].

For the purpose of five-fold cross validation, we partitioned the ratings for each individual into five folds of equal size. For each individual, we held out one fold as the testing set and used the remaining four folds as the training set. We repeated this process five times with each fold held out exactly once as the testing set and the remaining four folds used as the training set. With the ten datasets, this resulted in 50 experiments for each evaluation candidate.

6.1.1 Performance Metrics

Rating Prediction Accuracy

As proposed in [Herlocker et al., 2004], we use the mean absolute error (MAE) and the root of the mean squared error (RMSE) to measure the recommendation performance in terms of rating prediction accuracy.

MAE. The MAE is defined as the mean of the absolute difference between an individual i's provided rating r_i and the corresponding predicted rating

$\widehat{r_i}$ for all ratings $r \in R$. We denote the number of ratings as $n = |R|$. The MAE is defined as follows:

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |r_i - \widehat{r_i}| \tag{6.1}$$

RMSE. Similarly, the RMSE is the root of the mean squared error. The RMSE complements the MAE because it is more sensitive to large prediction errors rather than small prediction errors. In other words, the RMSE penalizes large prediction errors, whereas it rewards small prediction errors. Thus, it better represents individuals' perception of performance quality. The RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (r_i - \widehat{r_i})^2} \tag{6.2}$$

The rating prediction errors are normally distributed. Therefore, the absolute errors are folded normally distributed. For this reason, we used the non-parametric Wilcoxon signed-rank test for dependent samples to test pairwise the MAE and RMSE for significance. We set the significance level to $\alpha = 0.01$, which is stronger than the conventional significance level of $\alpha = 0.05$ [Stigler, 2008]. We set the significance level lower to provide stronger empirical evidence. We applied the Bonferroni correction to control for the family-wise error.

Relevance Filtering Quality

As complement to the rating prediction accuracy of the previous section, we measured the filtering quality of relevant products. For the evaluation, we use the MovieLens 100k dataset[1], which we present in Section 6.3. The movies in the MovieLens 100k dataset are rated on a 1-to-5 integer scale.

[1] `http://www.movielens.org`

But a binary scale is required to assess the filtering quality. Based on this reason, we define a product as relevant for a particular individual when his rating for the product is equals or higher relative to his rating median for the particular dataset.

In the following, we denote relevant products as *positives* (P) and irrelevant products as *negatives* (N). Furthermore, we denote recommended products which are relevant (i.e., positives) as *true-positives* (TP) and recommended products which are irrelevant (i.e., negatives) as *false-positives* (FP) respectively. Analogously, we denote not recommended products which are relevant as *false-negatives* (FN) respectively not recommended products which are irrelevant as *true-negatives* (TN). The true-positives and true-negatives are correctly classified products. The false-positives and false-negatives are wrongly classified products.

We measured the filtering quality by means of Precision and Recall, F_1-score, area under the ROC curve (AUC) and the Matthews correlation coefficient (MCC). The Precision and Recall are commonly used performance metrics to assess the quality of a binary classification (i.e., filter quality).

Precision. In our context, Precision shows the correctness of recommended products, the relation between true-positives (TP) and false-positives (FP) in particular. Hence, Precision is defined as follows:

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{6.3}$$

Recall. Recall, in turn, shows the quality of completeness of relevant products, the relation between true-positives (TP) and false-negatives (FN) in particular. Hence, Recall is defined as follows:

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{6.4}$$

The Precision and Recall values are based on a sequence of Bernoulli experiments and are thus binomial distributed. As stated by the central limit theorem, a binomial distribution approximates a normal distribution for large amount of Bernoulli experiments. Therefore, paired-samples t-test can be used to test for significance. We set the significance level α to $\alpha = 0.01$. We applied the Bonferroni correction to control for the family-wise error.

F_1-score. Both Precision and Recall are two dimensions of the quality of a binary classification. However, there is a trade-off between Precision and Recall. The reason being is that either value can be increased at the cost of the other one. For this reason, both Precision and Recall have to be taken into consideration to assess the filtering quality of relevant products. For this reason, we further computed the F_1-score, which accounts for the well-known trade-off between Precision and Recall and is commonly used. The F_1-score is defined as the harmonic mean of Precision and Recall. Hence, it is defined as:

$$F_1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6.5}$$

MCC. However, the F_1-score does not account for the true-negative rate, which is an additional quality criteria of a binary classification. For this reason, we further measured MCC, which assess the correlation of the observed classification of products by an individual and the predicted classification. The MCC is defined as:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{6.6}$$

AUC. Nevertheless, both the F_1-score and MCC do not show the effect of improving the Recall at the cost of the Precision. The receiver operating characteristics (ROC) curve, in turn, shows the marginal benefit of increasing the Recall increasing the false-positive rate. The false-positive rate is defined as $\frac{FP}{N}$, which is related to Precision. We calculate the AUC as a summary statistic of the ROC curve. The AUC can be interpreted as the probability that a classifier ranks a randomly chosen relevant product higher than a randomly chosen irrelevant product.

6.2 Candidates for Comparison

In the following, we present the candidates for comparison. Firstly, we present collaborative filtering methods which use our proposed methods from Part III to retrieve like-minded individuals based on their respective hypothesized preferences. Secondly, we present baseline collaborative filtering methods in Section 6.2.2. Lastly, we present content filtering-based methods in Section 6.2.3.

6.2.1 Hypothesis-Based Collaborative Filtering Candidates

In the following, we present our candidates which implement our proposed HCF. Firstly, we present our approaches which retrieve like-minded individuals based on the comparison of the similarity of hypothesized utilities. Finally, we present our approaches which retrieve like-minded individuals based on the comparison of the composition of individuals' hypothesized preferences.

Hypothesized Utility-Based Collaborative Filtering Implementations

The following candidates for comparison are collaborative filtering methods and are based on the general framework for performing collaborative filtering [Herlocker et al., 1999], which we described in Section 2.2.1. The candidates retrieve like-minded individuals based on the HU preference similarity framework we presented in Section 5.2.

HUkNN-J48corr. Like-minded individuals are retrieved based on *correlative predicted utility-based similarity*, which is described in Section 5.2.2. The individuals preferences are hypothesized with the machine learning algorithm J48. The preference similarity between two individuals is computed as the correlation of both hypothesized utilities for their unified rated products.

HUkNN-SVMcorr. Like-minded individuals are retrieved based on *correlative predicted utility-based similarity*, which is described in Section 5.2.2. The individuals preferences are hypothesized with the machine learning algorithm SVM. The preference similarity between two individuals is computed as the correlation of both hypothesized utilities for their unified rated products.

HUkNN-NBcorr. Like-minded individuals are retrieved based on *correlative predicted utility-based similarity*, which is described in Section 5.2.2. The individuals preferences are hypothesized with the machine learning algorithm Naïve Bayes. The preference similarity between two individuals is computed as the correlation of both hypothesized utilities for their unified rated products.

HUkNN-J48prob. Like-minded individuals are retrieved based on *probabilistic predicted utility-based similarity*, which is described in Section 5.2.3. The individuals preferences are hypothesized with the machine learning algorithm J48. The preference similarity between two individuals is computed as the probability that products of the unified rated products provide the same hypothesized utilities to both individuals.

HUkNN-NBprob. Like-minded individuals are retrieved based on *probabilistic predicted utility-based similarity*, which is described in Section 5.2.3. The individuals preferences are hypothesized with the machine learning algorithm Naïve Bayes. The preference similarity between two individuals is computed as the probability that products of the unified rated products provide the same hypothesized utilities to both individuals.

HUkNN-sJ48prob. Partially like-minded individuals are retrieved based on *probabilistic predicted utility-based semi-partial similarity*, which is described in Section 5.2.4. The individuals preferences are hypothesized with the machine learning algorithm J48. The semi-partial preference similarity of two individuals is computed as the probability that the second individual receives the same utility as the first individual from products which corresponds to the first individual's hypothesized preferences.

Hypothesis Composition-Based Collaborative Filtering

The following candidates for comparison are collaborative filtering methods and are based on the general framework for performing collaborative filtering [Herlocker et al., 1999], which we described in Section 2.2.1. The candidates retrieve like-minded individuals based on the HC preference similarity framework we presented in Section 5.3.

HCkNN-J48NoG. The individuals' preferences are hypothesized with the machine learning algorithm J48. The Jaccard similarity coefficient (see Section 5.3.1) and rating equality similarity (see Section 5.3.1) are used to compute the similarity of HPPs and build the partial preference similarity matrix S. The similarities of the partial preference matrix S are consolidated with the noisy-or gate (see Section 5.3.2).

HCkNN-J48Sup. The individuals preferences are hypothesized with the machine learning algorithm J48. The Jaccard similarity coefficient (see Section 5.3.1) and rating equality similarity (see Section 5.3.1) are used to compute the similarity of HPPs and build the partial preference similarity matrix S. The similarities of the partial preference matrix S are consolidated with the supremum norm (see Section 5.3.2).

HCkNN-NBSup. The individuals preferences are hypothesized with the machine learning algorithm Naïve Bayes. The Jaccard similarity coefficient (see Section 5.3.1) and rating equality similarity (see Section 5.3.1) are used to compute the similarity of HPPs and build the partial preference similarity matrix S. The similarities of the partial preference matrix S are consolidated with the supremum norm (see Section 5.3.2). Note that the supremum operator and the noisy-or operator produce the same results in the case of Naïve Bayes.

HCkNN-J48SSup. The individuals preferences are hypothesized with the machine learning algorithm J48. The semantic extension of the Jaccard similarity coefficient (see Section 5.3.1) and rating equality similarity (see Section 5.3.1) are used to compute the similarity of HPPs and build the partial preference similarity matrix S. The similarities of the partial preference matrix S are consolidated with the supremum norm (see Section 5.3.2).

HCkNN-NBSSup. The individuals preferences are hypothesized with the machine learning algorithm Naïve Bayes. The semantic extension of the Jaccard similarity coefficient (see Section 5.3.1) and rating equality similarity (see Section 5.3.1) are used to compute the similarity of HPPs and build the partial preference similarity matrix S. The similarities of the partial preference matrix S are consolidated with the supremum norm (see Section 5.3.2).

6.2.2 Baseline Collaborative Filtering Candidates

We benchmark our proposed methods against three different collaborative filtering methods.

WoC. Wisdom of Crowds is the most basic collaborative filtering method. It ranks the products by popularity and recommends the most popular products. A product's popularity is computed by the average of all received ratings.

PcorrkNN. This is the well-known collaborative filtering method proposed in [Herlocker et al., 1999]. We used Pearson' correlation on common rated items to retrieve like-minded individuals. In all our experiments, we set $k = 10$ for the kNN-based collaborative filtering method.

SVD. This is the well-known matrix factorization approach proposed in [Sarwar et al., 2000], which is based on single value decomposition.

6.2.3 Baseline Content Filtering Candidates

Besides the collaborative filtering methods presented in Section 6.2.2, we benchmark our proposed methods against four content filtering methods. These methods hypothesize individuals' preferences based on their provided ratings for products and the respective product features. For this purpose, each candidate uses a different machine learning algorithm. These hypothesized preferences are then used to predict the utility a product provides to individuals. Note that these methods are related to our HCF methods regarding the machine learning algorithms which are used in both cases to hypothesize individuals' preferences. We use the machine learning algorithms provided by the Weka library [Witten and Frank, 2005] for the following four content filtering methods.

J48. The decision tree induction algorithm J48 is used to hypothesize individuals' preferences. The J48 is the Java implementation of the C4.5.

SVM. The support vector machine is used to hypothesize individuals' preferences. We use SVM which is a support vector machine using a linear kernel. More precisely, we use SMO, which is the SVM implementation in the Weka library.

Naïve Bayes The Naïve Bayes classifier is used to hypothesized individuals' preferences.

Bayes Net The Bayes Net is used to hypothesize individuals' preferences.

6.3 Dataset

For the purpose of evaluation, we considered the MovieLens 100k dataset from MovieLens[2], which provides 100 000 ratings of 943 individuals about 1 682 movies. The MovieLens dataset is a quasi-benchmark due to its widely use for the evaluation of collaborative filtering [Lawrence and Urtasun, 2009] and is provided by GroupLens[3].

In the following, we present the properties of the dataset, specifically the ratings, the individuals and the movies. The ratings are discrete values on a 1-to-5 integer scale with a rating mean of 3.53, a rating standard deviation of 1.13 and a rating median of four. The non-uniform distribution of rating values suggests that individuals tend to rate what they prefer and prefer what they rate. Each individual provides at least 20 ratings and at most 737 ratings. The mean of the number of ratings per individual is 106.04 with a median of 65 and a standard deviation of 100.93. The distribution of number of ratings per individual indicates that most individuals provide a

[2]http://www.movielens.org
[3]http://www.grouplens.org

reasonable amount of ratings. Each movie is rated at least once and at most 583. The mean of the number of ratings per movie is 59.45 with a median of 27 and a standard deviation of 80.40. The distribution of number of ratings per movie indicates that many movies are rarely rated, which corresponds to the Long Tail.

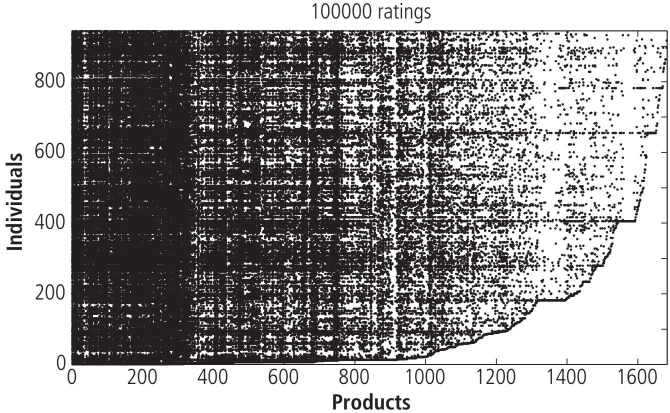
We model individuals' preferences in terms of movie properties. For this purpose, we consider exclusively the movies' genre information (e.g., action, drama), which ultimately results in simple preference models. The MovieLens dataset provides 18 different types of genres, which we list in Table C.1 in Appendix C.1. Every movie is related to at least one genre and at most six genres. The median number of genres per movie is two.

We use our movie ontology MO[4] to which we map the genres of the MovieLens dataset to describe the movies. We provide the mapping of the MovieLens's genres to the movie ontology MO's genres in Table C.1 in Appendix C.1. The movie ontology MO provides amongst other things a taxonomy of genres, which we use as semantic relation among genres to enable the semantic comparison of different preferences.

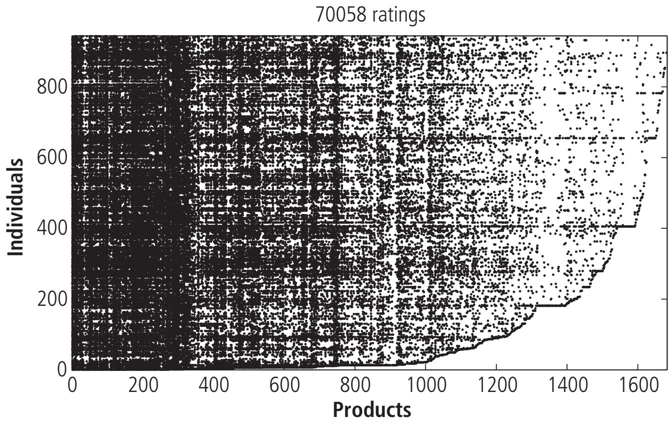
To evaluate the cold-start behavior, we derived nine datasets with different degree of rating sparsity from the MovieLens 100k dataset. For this purpose, we stepwise randomly removed 10% of the ratings per individual to derive nine datasets with different degree of rating sparsity. In contrast to randomly selecting a limited number of individuals from the original dataset, this method produces datasets, which have similar characteristics to the original dataset, thus being appropriate to analyze the cold-start behavior.

Figure 6.1 illustrates the increasing degree of rating sparsity starting from the original MovieLens 100k dataset presented in Figure 6.1a, to the dataset with 30%, 60% and 90% increased rating sparsity, which are presented in Figure 6.1b, Figure 6.1c and Figure 6.1d, respectively. As Figure 6.1

[4] http://www.movieontology.org

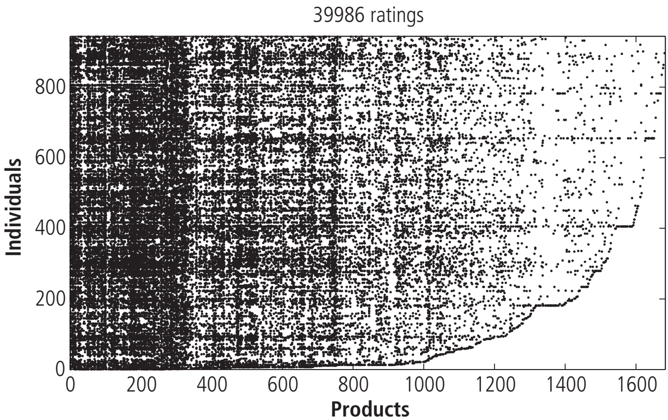


(a) Original MovieLens 100k dataset

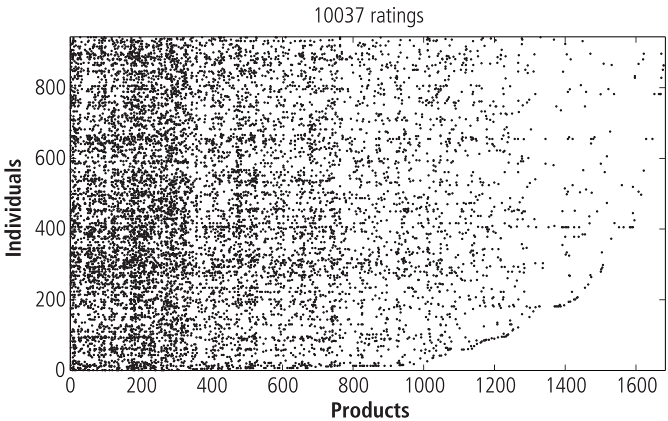


(b) Dataset with 30% ratings removed

Figure 6.1: MovieLens 100k dataset with stepwise increased degree of rating sparsity. A black dot indicates a rating of the particular individual for the particular product.



(c) Dataset with 60% of ratings removed



(d) Dataset with 90% of ratings removed

Figure 6.1: MovieLens dataset with stepwise increased degree of rating sparsity. A black dot indicates a rating of the particular individual for the particular product.

suggests, the characteristics of the derived datasets stays similar to the original MovieLens 100k dataset. We refer to Appendix C.2 for the complete list of illustrations of the sparse datasets.

6.4 Results and Discussion

In the following, we present the results of the rating prediction accuracy and the relevance filtering quality of our empirical study. The sparsity degree refers to the dataset with the accordant sparsity relative to the original MovieLens dataset (see Section 6.3). First, we present the results for the rating prediction accuracy in Section 6.4.1. Then, we present the results for the relevance filtering quality in Section 6.4.2.

6.4.1 Rating Prediction Accuracy

We evaluated the rating prediction accuracy of the methods in terms of MAE and RMSE. Table 6.1 and Table 6.2 present the MAE and RMSE values respectively of the evaluated methods. Figure 6.3 and Figure 6.4 provide a visual representation of the MAE and RMSE values respectively for the better comprehension of the results. For readability reason, both Figure 6.3 and Figure 6.4 provide a selection of the evaluated methods that are presented in Table 6.1 and Table 6.2 respectively.

As shown in Table 6.1 and Table 6.2 as well as in Figure 6.3 and Figure 6.4, all methods performed better regarding to MAE and RMSE the more ratings were provided by individuals in general. As a general pattern, the rating prediction accuracy of all methods remains relatively similar for datasets with low sparsity degree (0% to 60% sparsity). But the accuracy decreases super-proportional for datasets with high sparsity degree (70% to 90% sparsity). Furthermore, collaborative filtering methods (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-*

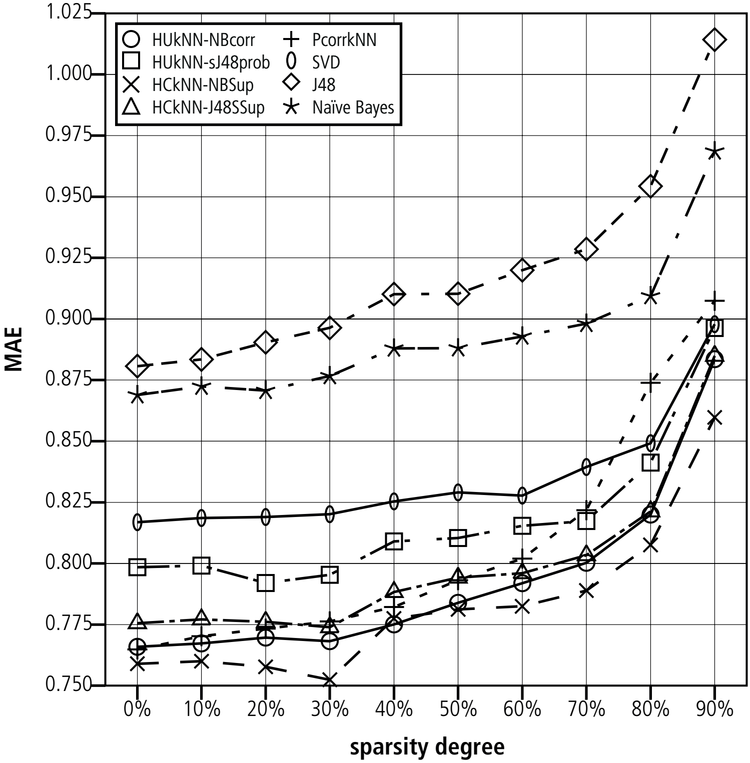


Figure 6.2: *HCkNN-NBSup*

Figure 6.3: Behavior of recommendation performance in terms of MAE with increasing degree of sparsity from 0% (original data set) to 90%

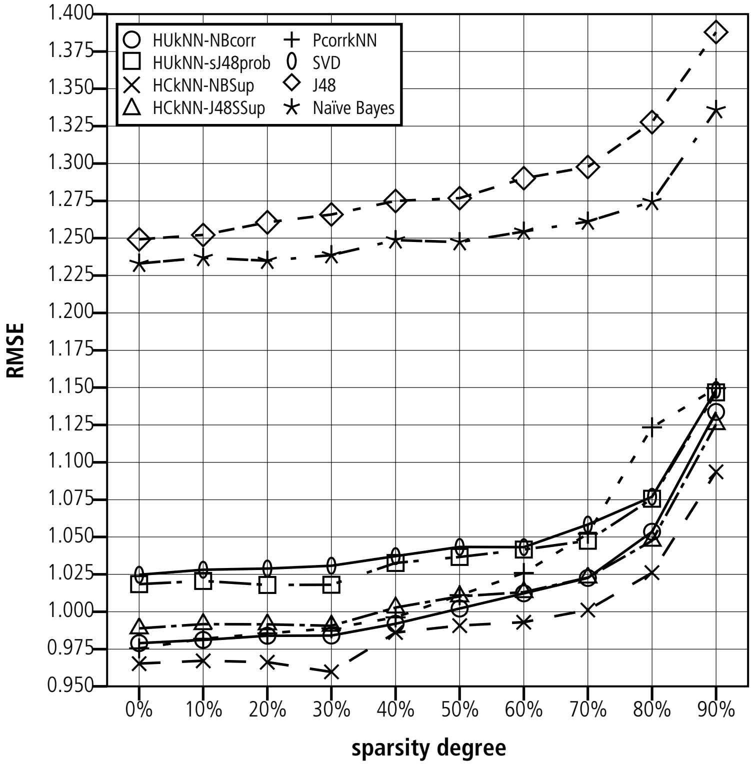


Figure 6.4: Behavior of recommendation performance in terms of RMSE with increasing degree of sparsity from 0% (original data set) to 90%.

Method	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
HUkNN-J48corr	0.803	0.805	0.808	0.807	0.816	0.824	0.830	0.839	0.859	0.900
HUkNN-NBcorr	0.766	0.767	0.770	0.768	**0.775**	0.784	0.792	0.800	0.820	0.884
HUkNN-SVMcorr	0.769	0.768	0.774	0.776	0.780	0.791	0.794	0.805	0.836	0.873
HUkNN-J48prob	0.789	0.788	0.790	0.790	0.796	0.800	0.802	0.807	0.829	0.876
HUkNN-NBprob	0.775	0.775	0.777	0.776	0.785	0.788	0.790	0.797	0.812	0.866
HUkNN-sJ48prob	0.799	0.799	0.792	0.795	0.809	0.810	0.815	0.817	0.841	0.896
HCkNN-J48NoG	0.786	0.788	0.791	0.797	0.794	0.801	0.803	0.820	0.845	0.922
HCkNN-J48Sup	0.786	0.788	0.791	0.796	0.794	0.801	0.804	0.819	0.844	0.923
HCkNN-NBSup	**0.759**	**0.760**	**0.758**	**0.752**	0.778	**0.781**	**0.783**	**0.789**	**0.808**	**0.860**
HCkNN-J48SSup	0.776	0.777	0.776	0.774	0.788	0.794	0.796	0.803	0.821	0.885
HCkNN-NBSSup	0.825	0.823	0.815	0.806	0.841	0.841	0.840	0.845	0.852	0.885
WoC	0.817	0.818	0.819	0.820	0.824	0.827	0.827	0.836	0.846	0.884
PcorrkNN	0.765	0.770	0.773	0.776	0.782	0.793	0.802	0.822	0.874	0.907
SVD	0.817	0.819	0.819	0.820	0.825	0.829	0.828	0.839	0.849	0.898
J48	0.881	0.884	0.891	0.896	0.910	0.910	0.920	0.929	0.954	1.014
Naïve Bayes	0.869	0.872	0.871	0.877	0.888	0.888	0.893	0.898	0.909	0.969
Bayes Net	0.887	0.890	0.890	0.898	0.909	0.913	0.918	0.920	0.939	0.999
SVM	0.854	0.860	0.864	0.874	0.882	0.888	0.900	0.909	0.934	1.010

Table 6.1: Cold-start behavior regarding MAE of rating predictions with increasing sparsity.

J48prob, *HUkNN-NBprob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup*, *HCkNN-NBSSup*, WoC, PcorrkNN, and SVD) significantly outperforms content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM).

The significantly best performing method[5] was *HCkNN-NBSup*, which uses Naïve Bayes to hypothesize individuals' preferences and retrieves like-minded individuals based on the comparison of the composition similarity of individuals' hypothesized preferences. Regarding to RMSE, *HCkNN-NBSup* outperformed all other methods. It outperformed all other methods regarding to MAE with respect to the datasets with the lowest sparsity

[5]tested with the Wilcoxon signed-rank test for dependent samples with significance level $\alpha = 0.01$ and the Bonferroni correction.

Method	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
HUkNN-J48corr	1.018	1.021	1.026	1.026	1.035	1.047	1.057	1.068	1.095	1.153
HUkNN-NBcorr	0.979	0.981	0.984	0.984	0.992	1.002	1.012	1.023	1.054	1.134
HUkNN-SVMcorr	0.983	0.983	0.990	0.992	0.998	1.011	1.015	1.030	1.070	1.128
HUkNN-J48prob	0.999	1.000	1.002	1.002	1.008	1.015	1.018	1.024	1.055	1.120
HUkNN-NBprob	0.983	0.984	0.987	0.986	0.995	1.000	1.004	1.012	1.034	1.103
HUkNN-sJ48prob	1.018	1.021	1.018	1.018	1.033	1.037	1.042	1.048	1.076	1.147
HCkNN-J48NoG	1.003	1.007	1.013	1.023	1.011	1.019	1.026	1.046	1.080	1.172
HCkNN-J48Sup	1.002	1.005	1.013	1.021	1.009	1.019	1.025	1.045	1.079	1.172
HCkNN-NBSup	**0.965**	**0.967**	**0.966**	**0.960**	**0.986**	**0.991**	**0.993**	**1.001**	**1.026**	**1.094**
HCkNN-J48SSup	0.989	0.992	0.992	0.991	1.003	1.010	1.013	1.023	1.048	1.126
HCkNN-NBSSup	1.030	1.029	1.020	1.009	1.049	1.048	1.050	1.055	1.068	1.111
WoC	1.025	1.026	1.028	1.029	1.034	1.038	1.040	1.051	1.067	1.119
PcorrkNN	0.975	0.982	0.986	0.989	0.996	1.011	1.026	1.052	1.123	1.150
SVD	1.025	1.028	1.029	1.031	1.037	1.043	1.043	1.058	1.077	1.148
J48	1.249	1.252	1.261	1.266	1.275	1.277	1.290	1.298	1.328	1.388
Naïve Bayes	1.233	1.237	1.235	1.239	1.249	1.247	1.255	1.261	1.274	1.336
Bayes Net	1.254	1.256	1.257	1.262	1.272	1.273	1.283	1.285	1.304	1.366
SVM	1.213	1.221	1.227	1.236	1.241	1.248	1.262	1.272	1.301	1.375

Table 6.2: Cold-start behavior regarding RMSE of rating predictions with increasing sparsity.

degree (0%) and highest sparsity degree (90%), and almost all other datasets.

Generally, incorporating Naïve Bayes to hypothesize individuals' preferences in HCF methods (i.e., *HUkNN-NBcorr*, *HUkNN-NBprob*, and *HCkNN-NBSup*) provided competitive performance relative to the baseline collaborative filtering methods PcorrkNN, SVD and WoC, in particular when the sparsity degree gets higher. PcorrkNN was the only baseline collaborative filtering method, which performed significantly better than *HUkNN-NBprob* and *HUkNN-NBcorr* for datasets, even though marginal so and for datasets with very low sparsity degree (0% to 10%).

In contrast, incorporating J48 to hypothesize individuals' preference in HCF methods (i.e., *HUkNN-J48corr*, *HUkNN-J48prob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup* and *HCkNN-J48SSup*) provided generally only significantly better performance relative to the baseline collaborative filtering methods PcorrkNN, SVD and WoC when the sparsity degree is high.

However, the comparison of our methods which are based on the HU preference similarity framework (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, and *HUkNN-NBprob*) and our methods based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-NBSSup*, and *HCkNN-J48SSup*) shows that neither method completely outperforms the other. The probabilistic-based method *HUkNN-NBprob* performed significantly better than *HUkNN-NBcorr* with high sparsity degree. The correlation-based method *HUkNN-NBcorr*, in turn, benefits more from additional ratings and therefore performs significantly better than *HUkNN-NBprob* with low sparsity degree.

The methods which are based on HU preference similarity framework performed generally better when incorporating Naïve Bayes relative to incorporating SVM or J48. Therefore, we recommend based on the results presented in Table 6.1 and Table 6.2 to favor Naïve Bayes over SVM and J48.

Partial preference similarity or rather semi-partial preference similarity did not provide more accurate rating predictions as the results of *HUkNN-sJ48prob* show. In fact, it provided less accurate rating predictions than methods considering the overall preference similarity between individuals. We think that the overall similarity has to be considered when retrieving like-minded individuals based on partial preference similarities. This open research question, however, is not part of our investigation and remains as further research opportunity.

The methods which are based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup* and *HCkNN-NBSSup*) perform generally well, in particular *HCkNN-NBSup*. Likewise to the case of HU preference similarity framework, the methods perform significantly better when using Naïve Bayes to hypothesize preferences instead of J48. The consolidation methods (i.e., supremum norm and noisy-or gate) provide nearly the same results. The noisy-or gate, however, requires more computational effort than the supremum norm. Hence, we recommend the supremum operator because of the less computational effort.

Exploiting the semantic similarity of HPPs in *HCkNN-J48SSup* provides a significant overall improvement over the similar method *HCkNN-J48Sup*. The relative improvement gets smaller the more ratings are available. In contrast, exploiting the semantic similarity of HPPs in *HCkNN-NBSSup* reduces the performance compared to the similar method *HCkNN-NBSup*, which is the best performing method compared to all others. Therefore, exploiting semantic similarity of partial preferences does not necessarily provide an improvement and can even reduce the performance.

The content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM) perform significantly worse compared to all other methods. Nevertheless, the results show that machine learning algorithms efficiently hypothesize individuals preferences. As indicated in Figure 6.3 and Figure 6.4, these methods efficiently hypothesize individuals' preferences by means of generalizing from few ratings. More ratings, however, provide little information and basically verify the hypothesized preferences.

6.4.2 Relevance Filtering Quality

We evaluated the relevance filtering quality of the methods regarding to Precision, Recall, F_1-score, MCC and AUC. In the following, we first present the results regarding to Precision followed by the results regarding to Recall. Next, we present the results regarding to F_1-score, MCC and AUC.

Precision

As shown in Table 6.3, all methods performed better regarding to Precision the more ratings were provided by individuals. As a general pattern, the Precision decreases super-proportional for datasets with high sparsity degree (70% to 90% sparsity), i.e., major cold-start situation. Furthermore, collaborative filtering methods (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, *HUkNN-NBprob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup*, *HCkNN-NBSSup*, WoC, PcorrkNN, and SVD) significantly outperform content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM). Figure 6.5 shows the Precision values for the better comprehension of the results. For readability reason, Figure 6.5 provides a selection of the evaluated methods that are presented in Table 6.3.

The significantly best performing method was *HCkNN-NBSup*, which uses Naïve Bayes to hypothesize individuals' preferences and retrieves like-minded individuals based on the comparison of the composition similarity of individuals' hypothesized preferences. In contrast to the performance presented in Section 6.4.1, the baseline collaborative filtering methods SVD and WoC significantly outperform *HCkNN-NBSup* for datasets with high sparsity degree (i.e., 80% and 90%).

More generally, incorporating Naïve Bayes to hypothesize individuals' preferences in HCF methods (i.e., *HCkNN-NBSup*, *HUkNN-NBprob*, and *HUkNN-NBcorr*) provided competitive performance relative to the baseline collaborative filtering methods PcorrkNN, SVD and WoC, particularly when the sparsity degree gets lower. PcorrkNN was the only baseline collaborative filtering method, which performed significantly worse than *HCkNN-NBSup*, *HUkNN-NBprob*, and *HUkNN-NBcorr* for almost all datasets. In contrast to the results in Section 6.4.1, SVD and WoC significantly outperformed our methods *HCkNN-NBSup*, *HUkNN-NBprob*, and *HUkNN-NBcorr*, even though for datasets with very high sparsity degree.

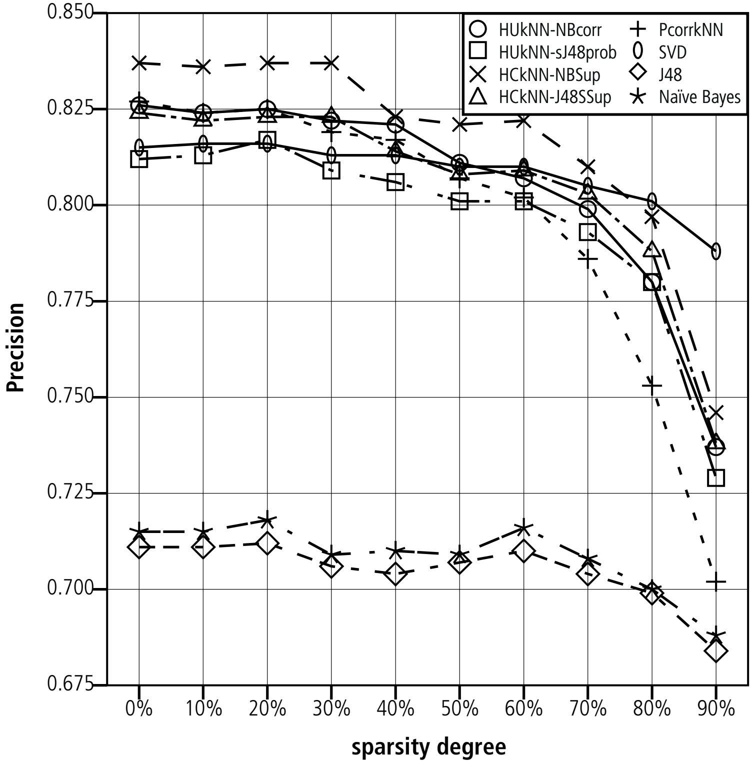


Figure 6.5: Precision with increasing degree of sparsity from 0% (original data set) to 90%.

Method	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
HUkNN-J48corr	0.804	0.799	0.800	0.795	0.790	0.781	0.777	0.767	0.745	0.720
HUkNN-NBcorr	0.826	0.824	0.825	0.822	0.821	0.811	0.807	0.799	0.780	0.737
HUkNN-SVMcorr	0.825	0.824	0.822	0.818	0.816	0.806	0.806	0.791	0.774	0.740
HUkNN-J48prob	0.820	0.820	0.818	0.815	0.814	0.811	0.813	0.801	0.786	0.740
HUkNN-NBprob	0.827	0.825	0.827	0.825	0.821	0.815	0.816	0.807	0.796	0.744
HUkNN-sJ48prob	0.812	0.813	0.817	0.809	0.806	0.801	0.801	0.793	0.780	0.729
HCkNN-J48NoG	0.819	0.817	0.818	0.813	0.813	0.807	0.805	0.795	0.779	0.728
HCkNN-J48Sup	0.820	0.819	0.819	0.813	0.814	0.808	0.807	0.795	0.780	0.728
HCkNN-NBSup	**0.837**	**0.836**	**0.837**	**0.837**	**0.823**	**0.821**	**0.822**	**0.810**	0.797	0.746
HCkNN-J48SSup	0.824	0.822	0.823	0.823	0.814	0.808	0.809	0.803	0.788	0.738
HCkNN-NBSSup	0.786	0.784	0.792	0.797	0.767	0.766	0.762	0.754	0.732	0.694
WoC	0.814	0.814	0.814	0.811	0.811	0.807	0.808	0.802	0.797	0.775
PcorrkNN	0.827	0.824	0.825	0.819	0.817	0.807	0.802	0.786	0.753	0.702
SVD	0.815	0.816	0.816	0.813	0.813	0.810	0.810	0.805	**0.801**	**0.788**
J48	0.711	0.711	0.712	0.706	0.704	0.707	0.710	0.704	0.699	0.684
Naïve Bayes	0.715	0.715	0.718	0.709	0.710	0.709	0.716	0.708	0.700	0.688
Bayes Net	0.718	0.718	0.721	0.713	0.713	0.713	0.720	0.712	0.706	0.691
SVM	0.710	0.711	0.715	0.706	0.708	0.710	0.717	0.711	0.705	0.693

Table 6.3: Cold-start behavior regarding Precision with increasing sparsity.

In contrast, incorporating J48 to hypothesize individuals' preference in HCF methods (i.e., *HUkNN-J48corr*, *HUkNN-J48prob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup* and *HCkNN-J48SSup*) provided with few exceptions significantly lower Precision performance relative to the baseline collaborative filtering methods PcorrkNN, SVD and WoC. This results suggests that J48 is not appropriate to provide competitive Precision of recommendations.

However, the comparison of our methods which are based on the HU preference similarity framework (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, and *HUkNN-NBprob*) and our methods based on the HC preference similarity framework (i.e., *HCkNN-*

J48NoG, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-NBSSup*, and *HCkNN-J48SSup*) shows that neither method completely outperforms the other. The probabilistic-based method *HUkNN-NBprob* performed significantly better than *HUkNN-NBcorr*, particularly when the sparsity degree gets higher.

The methods which are based on HU preference similarity framework performed generally better when incorporating Naïve Bayes relative to incorporating SVM or J48. Based on the results presented in Table 6.3 we favor Naïve Bayes over SVM and J48.

Partial preference similarity or rather semi-partial preference similarity did not provide more accurate rating predictions as the results of *HUkNN-sJ48prob* show. In fact, it provided less accurate rating predictions than methods considering the overall preference similarity between individuals. We think that the overall similarity has to be considered when retrieving like-minded individuals based on partial preference similarities. This open research question, however, is not part of our investigation and remains as further research opportunity.

The methods which are based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup* and *HCkNN-NBSSup*) perform generally well, *HCkNN-NBSup* in particular. Likewise to the case of the HU preference similarity framework, the methods perform significantly better when using Naïve Bayes to hypothesize preferences instead of using J48. The consolidation methods (i.e., supremum norm and noisy-or gate) provide nearly same results. The noisy-or gate, however, requires more computational effort than the supremum norm. Hence, we recommend the supremum operator because of the less computational effort.

Exploiting the semantic similarity of HPPs in *HCkNN-J48SSup* provides a significant overall improvement compared to the similar method *HCkNN-J48Sup*. In contrast, exploiting the semantic similarity of HPPs in *HCkNN-NBSSup* reduces the performance compared to the similar method *HCkNN-*

NBSup, which is the best performing method compared to all others. Therefore, exploiting semantic similarity of HPPs does not necessarily provide an improvement and can even reduce the performance.

The content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM) perform significantly worst compared to all other methods. Nevertheless, the results show that machine learning algorithms efficiently hypothesize individuals preferences. As the results in Table 6.3 show, these methods efficiently hypothesize individuals' preferences by means of generalizing from few ratings. More ratings, however, provide little information and basically verify the hypothesized preferences.

Recall

As shown in Table 6.4, the Recall of most methods remains relatively constant with the exception of J48, Naïve Bayes, Bayes Net, and SVM, which performed performed better the more ratings were provided by individuals in general. In contrast to the results in Section 6.4.1 and the results regarding to Precision, the content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM) perform significantly and substantially better compared to all other methods. Figure 6.6 provides a visual representation of the Recall values for the better comprehension of the results. For readability reason, Figure 6.6 provides a selection of the evaluated methods that are presented in Table 6.4.

The significantly best Recall was provided by the baseline content filtering methods, specifically Naïve Bayes and SVM. On the other hand, *HCkNN-NBSSup* provided the significantly and substantially lowest Recall. Exploiting the semantic similarity of HPPs in *HCkNN-J48SSup* and *HCkNN-NBSSup* provided generally lower Recall than their accordant *HCkNN-J48Sup* respectively *HCkNN-NBSup*, which do not consider semantic similarities of HPPs. Therefore, exploiting semantic similarity of HPPs generally reduces the Recall.

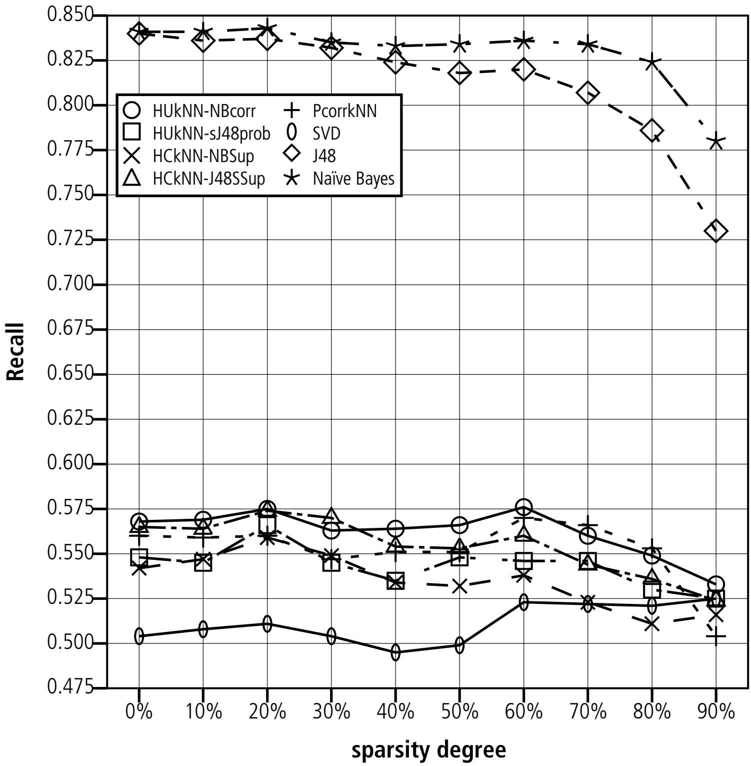


Figure 6.6: Recall with increasing degree of sparsity from 0% (original data set) to 90%.

Method	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
HUkNN-J48corr	0.524	0.525	0.531	0.526	0.525	0.526	0.540	0.528	0.526	0.530
HUkNN-NBcorr	0.568	0.569	0.575	0.563	0.564	0.566	0.576	0.560	0.549	0.533
HUkNN-SVMcorr	0.571	0.570	0.570	0.566	0.566	0.562	0.571	0.559	0.547	0.538
HUkNN-J48prob	0.522	0.523	0.527	0.520	0.523	0.524	0.539	0.530	0.520	0.516
HUkNN-NBprob	0.522	0.525	0.530	0.519	0.519	0.519	0.534	0.521	0.520	0.513
HUkNN-sJ48prob	0.548	0.545	0.566	0.545	0.535	0.548	0.546	0.546	0.530	0.525
HCkNN-J48NoG	0.569	0.566	0.579	0.571	0.558	0.557	0.566	0.545	0.535	0.527
HCkNN-J48Sup	0.559	0.559	0.573	0.567	0.547	0.546	0.560	0.542	0.533	0.526
HCkNN-NBSup	0.542	0.547	0.559	0.549	0.534	0.532	0.538	0.523	0.511	0.516
HCkNN-J48SSup	0.565	0.564	0.574	0.570	0.554	0.553	0.560	0.544	0.536	0.524
HCkNN-NBSSup	0.449	0.453	0.486	0.479	0.429	0.427	0.468	0.422	0.445	0.478
WoC	0.508	0.513	0.517	0.510	0.502	0.508	0.535	0.537	0.541	0.559
PcorrkNN	0.560	0.559	0.560	0.547	0.551	0.551	0.570	0.566	0.553	0.504
SVD	0.504	0.508	0.511	0.504	0.495	0.499	0.523	0.522	0.521	0.525
J48	0.840	0.836	0.837	0.832	0.824	0.818	0.820	0.807	0.786	0.730
Naïve Bayes	0.841	0.841	0.843	0.835	**0.833**	**0.834**	**0.836**	**0.834**	**0.824**	**0.780**
Bayes Net	0.822	0.821	0.823	0.812	0.810	0.808	0.810	0.803	0.793	0.752
SVM	**0.857**	**0.853**	**0.851**	**0.839**	0.833	0.828	0.824	0.810	0.789	0.733

Table 6.4: Cold-start behavior regarding Recall with increasing sparsity.

The comparison of our methods which are based on the HU preference similarity framework (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, and *HUkNN-NBprob*) and our methods based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-NBSSup*, and *HCkNN-J48SSup*) shows that neither method completely outperforms the other. The probabilistic-based method *HUkNN-NBprob* performed significantly better than *HUkNN-NBcorr*, particularly when the sparsity degree gets higher.

The methods which are based on the HU preference similarity framework performed generally better when computing the the preference similarity based on the correlation of hypothesized utilities for some products. Based on the results presented in Table 6.4 we favor the correlative similarity presented in Section 5.2.2 over the probabilistic similarity presented in Section 5.2.3.

Partial preference similarity or rather semi-partial preference similarity provided higher Recall than our methods incorporating the same machine learning algorithm to hypothesized preferences. Nevertheless, compared to other collaborative filtering methods, the Recall remains lower.

The methods which are based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, and *HCkNN-J48SSup*) performed competitive when the sparsity degree is low. The HC methods performed better when incorporating J48 than incorporating Naïve Bayes. This results suggest that J48 should be favored over Naïve Bayes when Recall is more important than Precision. Furthermore, the consolidation method noisy-or gate provides significantly higher Recall results than the supremum norm, even though marginal. Nonetheless, we recommend the supremum operator because of the less computational effort.

F_1-score, MCC and AUC

As discussed in Section 6.1.1, there is a trade-off between Precision and Recall. To provide a final conclusion about the overall filtering quality, we evaluated the methods in terms of F_1-score, MCC and AUC.

The F_1-score values presented in Table 6.5 suggest that the baseline content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM) provide an overall better filtering quality than the collaborative filtering methods (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, *HUkNN-NBprob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup*, *HCkNN-NBSSup*, WoC, PcorrkNN, and SVD). Nonetheless, we argue that Precision is more important than Recall to help individuals in choosing relevant products because of information overload and overchoice. In other words, a narrowed set of relevant prod-

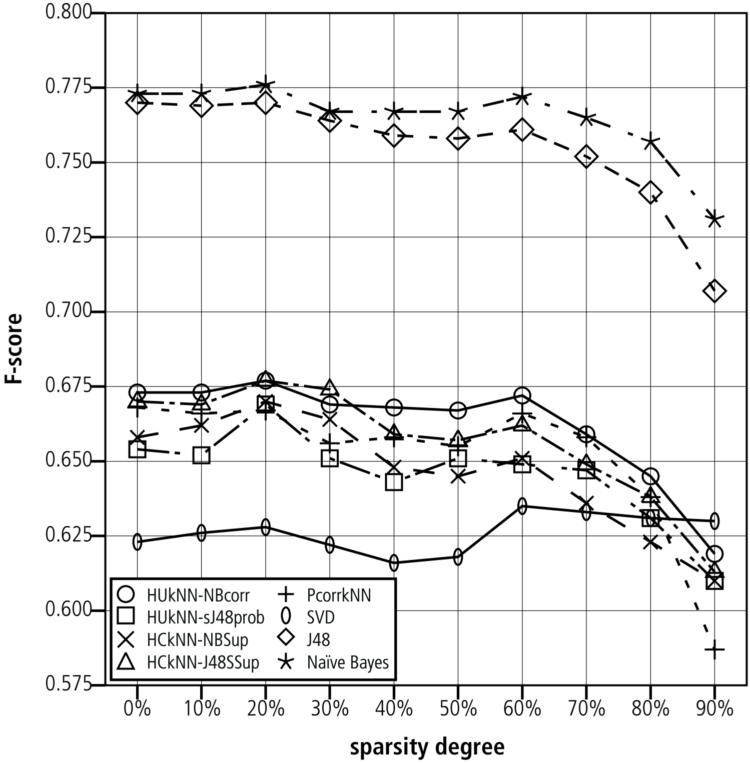


Figure 6.7: F_1-score with increasing degree of sparsity from 0% (original data set) to 90%.

Method	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
HUkNN-J48corr	0.634	0.634	0.638	0.633	0.631	0.629	0.637	0.625	0.617	0.610
HUkNN-NBcorr	0.673	0.673	0.677	0.669	0.668	0.667	0.672	0.659	0.645	0.619
HUkNN-SVMcorr	0.675	0.674	0.673	0.669	0.669	0.663	0.669	0.655	0.641	0.623
HUkNN-J48prob	0.638	0.639	0.641	0.635	0.637	0.637	0.648	0.638	0.625	0.608
HUkNN-NBprob	0.640	0.642	0.646	0.637	0.636	0.634	0.646	0.633	0.629	0.608
HUkNN-sJ48prob	0.654	0.652	0.669	0.651	0.643	0.651	0.649	0.647	0.631	0.610
HCkNN-J48NoG	0.671	0.669	0.678	0.671	0.662	0.659	0.665	0.647	0.634	0.612
HCkNN-J48Sup	0.665	0.664	0.674	0.668	0.654	0.651	0.661	0.644	0.633	0.611
HCkNN-NBSup	0.658	0.662	0.670	0.664	0.648	0.645	0.651	0.636	0.623	0.610
HCkNN-J48SSup	0.670	0.669	0.677	0.674	0.659	0.657	0.662	0.649	0.638	0.613
HCkNN-NBSSup	0.571	0.574	0.602	0.598	0.550	0.548	0.579	0.541	0.554	0.566
WoC	0.625	0.629	0.632	0.626	0.621	0.623	0.644	0.643	0.644	0.649
PcorrkNN	0.668	0.666	0.667	0.656	0.658	0.655	0.666	0.658	0.638	0.587
SVD	0.623	0.626	0.628	0.622	0.616	0.618	0.635	0.633	0.631	0.630
J48	0.770	0.769	0.770	0.764	0.759	0.758	0.761	0.752	0.740	0.707
Naïve Bayes	0.773	0.773	0.776	**0.767**	**0.767**	**0.767**	**0.772**	**0.765**	**0.757**	**0.731**
Bayes Net	0.766	0.766	0.769	0.759	0.758	0.757	0.762	0.755	0.747	0.720
SVM	**0.777**	**0.775**	**0.777**	**0.767**	0.766	0.764	0.767	0.757	0.745	0.712

Table 6.5: Cold-start behavior regarding F_1-score with increasing sparsity.

ucts is more important than a huge set in which individuals have to find relevant products. Figure 6.7 provides a visual representation of the F_1-score values for the better comprehension of the results. For readability reason, Figure 6.7 provides a selection of the evaluated methods that are presented in Table 6.5.

In contrast to the F_1-score values presented in Table 6.5, the MCC and AUC values presented in Table 6.6 and Table 6.7 respectively provide a more appropriate picture of the overall filter quality. For instance, in terms of MCC and AUC, the content filter methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM) provide the least filtering quality. Note that this result corresponds to also the rating prediction accuracy we discussed in Section 6.4.1.

Method	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
HUkNN-J48corr	0.232	0.224	0.222	0.221	0.210	0.193	0.179	0.166	0.126	0.059
HUkNN-NBcorr	0.289	0.285	0.283	0.283	**0.279**	0.262	0.248	0.235	0.198	0.096
HUkNN-SVMcorr	0.288	0.286	0.276	0.277	0.272	0.251	0.244	0.221	0.186	0.102
HUkNN-J48prob	0.258	0.257	0.249	0.252	0.250	0.244	0.242	0.228	0.198	0.100
HUkNN-NBprob	0.269	0.267	0.267	0.267	0.260	0.247	0.245	0.233	0.216	0.107
HUkNN-sJ48prob	0.255	0.256	0.266	0.253	0.242	0.237	0.225	0.219	0.191	0.079
HCkNN-J48NoG	0.278	0.271	0.273	0.271	0.263	0.250	0.240	0.223	0.191	0.077
HCkNN-J48Sup	0.275	0.271	0.272	0.269	0.259	0.247	0.240	0.221	0.192	0.076
HCkNN-NBSup	**0.294**	**0.295**	**0.296**	**0.302**	0.270	**0.263**	**0.258**	**0.239**	0.215	0.111
HCkNN-J48SSup	0.284	0.279	0.280	0.287	0.263	0.251	0.244	0.236	0.208	0.096
HCkNN-NBSSup	0.177	0.174	0.192	0.207	0.142	0.139	0.131	0.117	0.086	0.003
WoC	0.242	0.244	0.239	0.242	0.237	0.231	0.233	0.231	**0.225**	0.179
PcorrkNN	0.287	0.280	0.277	0.270	0.267	0.248	0.235	0.214	0.148	0.020
SVD	0.242	0.245	0.240	0.243	0.237	0.232	0.231	0.230	0.224	**0.191**
J48	0.120	0.120	0.108	0.110	0.092	0.095	0.083	0.077	0.057	-0.030
Naïve Bayes	0.139	0.134	0.132	0.122	0.117	0.111	0.111	0.098	0.070	-0.018
Bayes Net	0.141	0.139	0.136	0.128	0.121	0.115	0.117	0.108	0.085	-0.008
SVM	0.126	0.122	0.122	0.110	0.111	0.109	0.110	0.106	0.080	0.002

Table 6.6: Cold-start behavior regarding MCC with increasing sparsity.

Furthermore, we want to emphasize that the relative similarity of MCC and AUC values of any method is very high as it is shown in Table 6.6 and Table 6.7. In other words, the correlation of MCC and AUC with respect to a particular method is very high.

As shown in Table 6.1 and Table 6.2 as well as in Figure 6.3 and Figure 6.4, all methods performed better regarding to MAE and RMSE the more ratings were provided by individuals in general. As a general pattern, the rating prediction accuracy of all methods remains relatively similar for datasets with low sparsity degree (0% to 60% sparsity). But the accuracy decreases super-proportional for datasets with high sparsity degree (70% to 90% sparsity). Furthermore, collaborative filtering

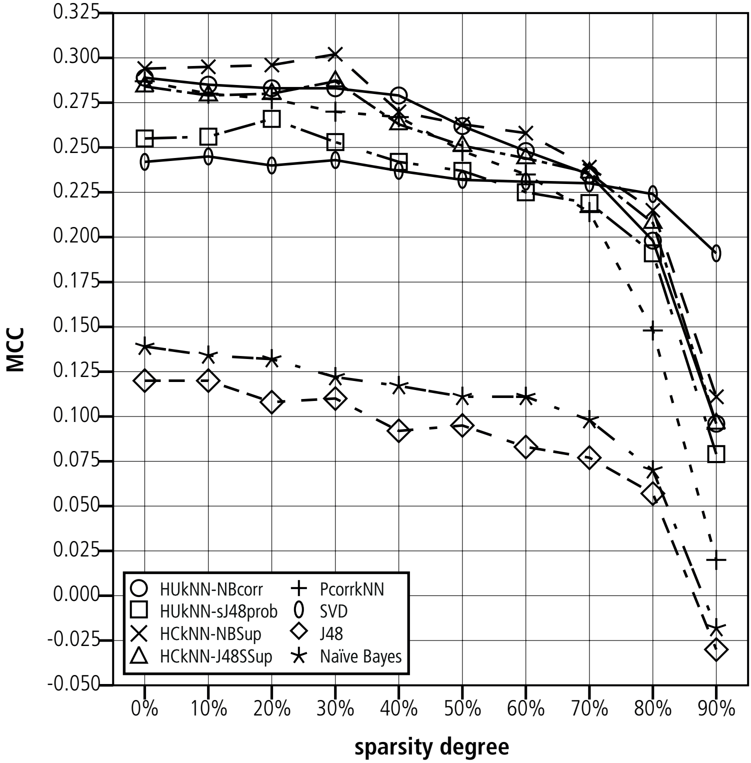


Figure 6.8: MCC with increasing degree of sparsity from 0% (original data set) to 90%.

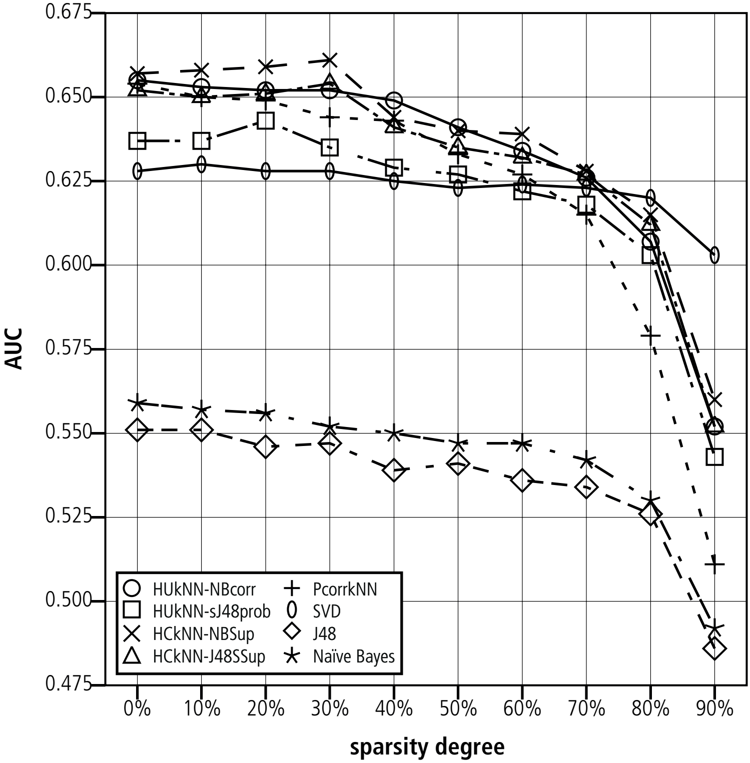


Figure 6.9: AUC with increasing degree of sparsity from 0% (original data set) to 90%.

Method	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
HUkNN-J48corr	0.624	0.620	0.619	0.618	0.613	0.604	0.597	0.589	0.568	0.532
HUkNN-NBcorr	0.655	0.653	0.652	0.652	**0.649**	**0.641**	0.634	0.626	0.607	0.552
HUkNN-SVMcorr	0.655	0.653	0.649	0.648	0.646	0.635	0.632	0.619	0.600	0.555
HUkNN-J48prob	0.638	0.637	0.634	0.634	0.633	0.630	0.630	0.622	0.606	0.554
HUkNN-NBprob	0.643	0.642	0.643	0.642	0.638	0.632	0.632	0.624	0.616	0.558
HUkNN-sJ48prob	0.637	0.637	0.643	0.635	0.629	0.627	0.622	0.618	0.603	0.543
HCkNN-J48NoG	0.649	0.645	0.647	0.645	0.641	0.634	0.630	0.620	0.603	0.542
HCkNN-J48Sup	0.647	0.646	0.647	0.644	0.639	0.632	0.630	0.618	0.603	0.541
HCkNN-NBSup	**0.657**	**0.658**	**0.659**	**0.661**	0.644	0.640	**0.639**	**0.628**	0.615	0.560
HCkNN-J48SSup	0.652	0.650	0.651	0.654	0.641	0.635	0.632	0.627	0.612	0.552
HCkNN-NBSSup	0.593	0.591	0.602	0.609	0.574	0.572	0.570	0.561	0.545	0.502
WoC	0.628	0.630	0.628	0.628	0.626	0.623	0.626	0.624	**0.621**	0.597
PcorrkNN	0.654	0.650	0.649	0.644	0.643	0.633	0.627	0.615	0.579	0.511
SVD	0.628	0.630	0.628	0.628	0.625	0.623	0.624	0.623	0.620	**0.603**
J48	0.551	0.551	0.546	0.547	0.539	0.541	0.536	0.534	0.526	0.486
Naïve Bayes	0.559	0.557	0.556	0.552	0.550	0.547	0.547	0.542	0.530	0.492
Bayes Net	0.563	0.562	0.560	0.557	0.554	0.552	0.553	0.549	0.539	0.496
SVM	0.552	0.550	0.550	0.546	0.547	0.547	0.548	0.547	0.536	0.501

Table 6.7: Cold-start behavior regarding AUC with increasing sparsity.

methods (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, *HUkNN-NBprob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup*, *HCkNN-NBSSup*, WoC, PcorrkNN, and SVD) significantly outperform content filtering methods (i.e., J48, Naïve Bayes, Bayes Net, and SVM).

The significantly best performing method was *HCkNN-NBSup*, which uses Naïve Bayes to hypothesize individuals' preferences and retrieves like-minded individuals based on the comparison of the composition similarity of individuals' hypothesized preferences. Regarding to MCC and AUC, *HCkNN-NBSup* outperforms the other methods with respect to almost all other datasets. Only SVD and WoC perform better when the sparsity degree is very high (80% to 90%), i.e., major cold-start situation. Both baseline methods do not retrieve like-minded individuals, however. Thus, *HCkNN-NBSup* is the best performing method which retrieves like-minded individuals to filter relevant products.

Generally, incorporating Naïve Bayes to hypothesize individuals' preferences in HCF methods (i.e., *HUkNN-NBcorr*, *HUkNN-NBprob*, and *HCkNN-NBSup*) provided competitive performance relative to the baseline collaborative filtering methods PcorrkNN, SVD and WoC. More precisely, *HUkNN-NBcorr*, *HUkNN-NBprob* and *HCkNN-NBSup* perform better than PcorrkNN when the sparsity degree gets higher. In contrast, these methods perform worse than SVD and WoC when the sparsity degree gets higher.

Similarly, incorporating J48 to hypothesize individuals' preferences in HCF methods (i.e., *HUkNN-J48corr*, *HUkNN-J48prob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup* and *HCkNN-J48SSup*) provides better performance than PcorrkNN when the sparsity degree gets higher. In turn, these methods perform worse than SVD and WoC when the sparsity degree is high, i.e., major cold-start situation. In conclusion, incorporating Naïve Bayes provides better results.

However, the comparison of our methods which are based on the HU preference similarity framework (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, and *HUkNN-NBprob*) and our methods based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-NBSSup*, and *HCkNN-J48SSup*) shows that neither method completely outperforms the other. The correlative-based methods *HUkNN-NBcorr* and *HUkNN-SVMcorr* performed significantly better than *HUkNN-NBprob* and *HUkNN-J48prob*, particularly when the sparsity degree gets lower. In contrast, *HUkNN-J48corr* performs significantly worse than the other correlative-based or probabilistic-based methods.

The methods which are based on the HU preference similarity framework performed generally better when incorporating Naïve Bayes or SVM relative to incorporating J48. Based on the results presented in Table 6.6 and Table 6.7 we favor Naïve Bayes and SVM over J48.

HUkNN-sJ48prob, which considers partial preference similarity or rather semi-partial preference similarity, did not perform significantly better than similar probabilistic-based methods (i.e., *HUkNN-J48prob* and *HUkNN-NBprob*). In fact, it provided less accurate rating predictions than methods considering the overall preference similarity between individuals. *HUkNN-sJ48prob* significantly outperformed *HUkNN-J48corr*, however.

The methods which are based on the HC preference similarity framework (i.e., *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup* and *HCkNN-NBSSup*) perform generally well, *HCkNN-NBSup* in particular. Likewise to the case of the HU preference similarity framework, the methods perform significantly better when using Naïve Bayes to hypothesize preferences instead of J48. The consolidation methods (i.e., supremum norm and noisy-or gate) provide nearly the same results. The noisy-or gate, however, requires more computational effort than the supremum norm. Hence, we recommend the supremum operator because of the less computational effort.

Exploiting the semantic similarity of HPPs in *HCkNN-J48SSup* provides a significant overall improvement compared to the similar method *HCkNN-J48Sup*. The relative improvement remains relative constant. In contrast, the exploiting the semantic similarity of HPPs in *HCkNN-NBSSup* reduces the performance compared to the similar method *HCkNN-NBSup*, which is the best performing method compared to all others. Therefore, exploiting semantic similarity of partial preferences does not necessarily provide an improvement and can even reduce the performance.

6.5 Information Theoretic Reflection of Hypothesized Preferences versus Product Ratings

Traditionally in collaborative filtering, the individual i's preferences are represented as the individual i's product rating vector $R_{i\cdot}$. The product rating vector $R_{i\cdot}$ approximates the individual i's preferences $u^i(g)$ proportionally to the number of ratings $|G_i|$ provided by individual i. The reason is that product ratings are independent from each other. In other words, the approximation rate with every additional product rating r_{ig} remains constantly $\frac{1}{m}$.

In contrast, the approximation rate of individual i's preferences with machine learning is characterized by a sigmoid function. This means that the approximation rate increases super-proportionally in the beginning, but towards the end, it converges to 0. The reason for this behavior is that every product rating r_{ig} is related also to all the product's properties. Since products share similar properties, product ratings are interrelated. Based on this, machine learning algorithms are able to generalize from product ratings to individual i's preferences. Towards the end, however, additional ratings mostly reinforce the hypothesized preferences and do not further contribute to the accuracy of the hypothesized preferences. Referring to Corollary 1, hypothesized preferences approximate preferences as much as the defined hypothesis space, which is a human-designers choice.

However, in combination with collaborative filtering, both types of preference models get interrelated to other preference models such that an individual's preferences get approximated super-proportionally. The evaluation results provide empirical evidence that comparing the similarity of hypothesized preferences to retrieve like-minded outperforms the comparison of the similarity of ratings for common rated products.

6.6 Acceptance of Hypotheses

We evaluated our methods and compared them to baseline methods. We measured their performance in terms of rating prediction accuracy and relevance filtering quality. In the following, we discuss the acceptance of the following hypotheses.

Hypothesized utility-based preference similarity hypothesis (H3.1). With reference to the empirical study on rating prediction accuracy (see Section 6.4.1), retrieving like-minded individuals based on the HU preference similarity framework outperforms baseline methods, especially when the sparsity degree gets higher. Furthermore, this method provides similar performance as retrieving like-minded individuals based on ratings for common rated products.

Regarding the relevance filtering quality, the methods based on HU preference similarity framework outperforms the similar baseline method PcorrkNN.

Since the methods based on our HU preference similarity framework outperforms other baseline methods, especially in terms of rating prediction accuracy, we conclude that it is appropriate to retrieve like-minded individuals. In fact, it may even better retrieve like-minded individuals.

Therefore, we accept H3.1.

Hypothesis composition-based preference similarity hypothesis (H3.2). Regarding to rating prediction accuracy, retrieving like-minded individuals based on the HC preference similarity framework outperforms baseline methods, particularly when the sparsity degree gets higher. Furthermore, this method provides similar or better performance than retrieving like-minded individuals based on ratings for common rated products. Especially, *HCkNN-NBSup* outperforms all other methods in almost any case.

Regarding to the relevance filtering quality, the methods based on HC preference similarity framework outperforms baseline in almost any case, especially *HCkNN-NBSup*

Since our proposed HC preference similarity framework outperforms other baseline methods, PcorrkNN in particular, we conclude that it is appropriate to retrieve like-minded individuals. In fact, it retrieves better like-minded individuals, especially *HCkNN-NBSup*

Therefore, we accept H3.2.

Hypothesized partial preference similarity hypothesis (H3.3). With regards to rating prediction accuracy, considering partial preferences or rather semi-partial preferences to retrieve like-minded individuals does in some cases outperform baseline collaborative filtering methods, especially in cold-start situations.

Regarding to the relevance filtering quality, the method in *HUkNN-sJ48prob*, which considers partial preference similarity generally did not outperform baseline collaborative filtering methods. For that reason, we conclude that *HUkNN-sJ48prob* does not retrieve like-minded individuals.

Considering rating prediction accuracy and relevance filtering quality, we conclude that considering partial preference similarity or rather semi-partial preference similarity is not appropriate to retrieve like-minded individuals. Nevertheless, we think that this method may be appropriate when considering overall similarity.

However, we do not accept H3.3.

Cold-start mitigation hypothesis (H3.4). Regarding to rating prediction accuracy, retrieving like-minded individuals by comparing the similarity of hypothesized preferences significantly and substantially outperformed baseline methods in cold-start situations, especially *HCkNN-NBSup*.

Regarding to the relevance filtering quality, our methods generally outperformed PcorrkNN in cold-start situations. However, the baseline collaborative filtering methods SVD and WoC performed significantly best in cold-start situations. Nonetheless, both baseline methods do not retrieve like-minded individuals to filter products. Thus, we argue that our methods perform significantly best in cold-start situations by means of retrieving like-minded individuals.

We conclude that it mitigates the cold-start problem since the retrieval of like-minded individuals based on the comparison of the similarity of hypothesized preferences outperforms the relevant baseline methods.

Therefore, we accept H3.4.

6.7 Summary

In this chapter, we conducted an empirical study and provided empirical evidence about the superiority of both presented algorithmic frameworks. In the next chapter, we use the research methodology of grounded theory to scrutinize the empirical results of this chapter to understand explain the cold-start behavior of HCF.

7

Analysis

WHILST performance metrics allow for comparing different methods they do not help us to understand the phenomenon that *hypothesis-based collaborative filtering* (HCF) provides better recommendations than other methods. For this reason, we conduct an explanatory study using grounded theory methodology as described in [Corbin and Strauss, 2008] to scrutinize the empirical results from Chapter 6 to understand and explain this phenomenon. More precisely, we compare the cold-start behavior of collaborative filtering retrieving like-minded individuals based on the comparison of hypothesized preferences and the comparison of ratings for common rated products. Based on this grounded theory, we verify Hypothesis H4 (rating predicate hypothesis).

In the following, we first present the employed research methodology grounded theory and the systematical analysis of the data in Section 7.1. Subsequently, we present and discuss the grounded hypotheses which explain the phenomenon in Section 7.2 followed by the consolidation of the grounded theory in Section 7.3. Finally, we validate the theory in Section 7.4 and discuss the acceptance of the Hypothesis H4 in Section 7.5.

7.1 Method

In the following, we explain the employed research methodology grounded theory in Section 7.1.1. Then, we describe the data collection and analysis in Section 7.1.2 and Section 7.1.3, respectively.

7.1.1 Grounded Theory

For our study, we use grounded theory methodology as described by Corbin and Strauss [Corbin and Strauss, 2008]. Grounded theory is a qualitative research approach for looking systematically at data aiming at the generation of theory grounded in empirical data. The three elements in grounded theory are data collection, data analysis and theory development, which are repeated until a phenomenon to be researched can be explained.

Grounded theory is well suited for our explanatory study because it proposes to start from a general research question and refining the question as well as the analysis of data. For example, we binned individuals and products with respect to rating predicates to ensure the verification of formulated hypotheses. Furthermore, grounded theory is appropriate to formulate a theory grounded in quantitative data [Glaser and Strauss, 1967]. In our explanatory study, we collect and analyze quantitative data from our conducted empirical study presented in Chapter 6.

We employe open coding to analyze the data. More precisely, we enrich the data with additional data regarding the relation between individuals, products and ratings. Subsequently, we perform axial coding and link this codes to categories (e.g., factors, consequences) which are usually found in grounded theory study.

7.1.2 Data Collection

From the empirical study presented in Chapter 6, we collected the individuals, the products and the ratings an individual assigns to a product. Additionally, we collected the predicted ratings of the evaluated methods, specifically HCF methods described in Section 6.2.1 (i.e., *HUkNN-J48corr*, *HUkNN-NBcorr*, *HUkNN-SVMcorr*, *HUkNN-J48prob*, *HUkNN-NBprob*, *HUkNN-sJ48prob*, *HCkNN-J48NoG*, *HCkNN-J48Sup*, *HCkNN-NBSup*, *HCkNN-J48SSup*, *HCkNN-NBSSup*) and the baseline collaborative filtering method PcorrkNN described in Section 6.2.2. Besides collecting their performance in terms of MAE, we collected the performance difference between HCF methods and the baseline collaborative filtering method PcorrkNN by means of the difference of the respective MAEs.

Furthermore, we aggregated and collected additional information from the empirical data. This information can be classified into individual properties, product properties and recommendation properties and are presented subsequently.

Individual properties

Effort. The number of ratings an individual provides corresponds to the individual's effort. The higher the individual's effort, the more is known about the individual's entire preferences. We assume that individuals with high effort are less affected by the cold-start problem than individuals with low effort.

Attitude. The mean of ratings an individual provides corresponds to the individual's attitude. The attitude represents an individual's degree of like or dislike for products in general. The higher or lower the individual's attitude, the more respectively less satisfaction an individual receives from products in general.

Selectivity. The standard deviation of ratings an individual provides corresponds to the individual's selectivity. The lower the selectivity, the more similar is the satisfaction received from different products.

Product properties

Visibility. The number of ratings a product receives corresponds to the product's visibility. The higher the visibility of a product, the more likely like-minded individuals have rated this product. Therefore, the accuracy of predicted ratings is more dependent on the precision of finding like-minded individuals.

Popularity. The mean of ratings a product receives corresponds to the product's popularity. The higher the popularity of a product, the more likely individuals like this product.

Polarization. The standard deviation of ratings a product receives corresponds to the product's polarization. The higher the polarization of a product, the more important are personalized recommendations.

Recommendation properties

MAE. The performance of a method is measured in terms of MAE. The smaller the MAE the higher the performance. More precisely, the higher the performance the closer the predicted ratings to the individuals' rating in general.

ΔMAE PcorrkNN. The difference of MAE values of a method of comparison and PcorrkNN corresponds to ΔMAE PcorrkNN. A positive ΔMAE PcorrkNN means that the method of comparison is inferior to PcorrkNN. In turn, a negative ΔMAE PcorrkNN means that the method of comparison is superior to PcorrkNN.

7.1.3 Data Analysis

For the analysis, we use the properties presented in Section 7.1.2 as codes. Since the values of these properties have an excessive variance and therefore are difficult to interpret, we bin individuals and products to 20-quantiles regarding to these properties to generalize from these values and allow for an interpretation. In other words, we create codes expressing the degree of an individual's and a product's property repspectively on a 1-to-20 scale. In a preliminary analysis, we compared 100-quantiles, 50-quantiles, 25-quantiles, 20-quantiles, 10-quantiles and 5-quantiles. We found 20-quantiles as the most appropriate to generalize from these values without biasing the data.

We use axial coding to associate codes to categories which are prescribed by grounded theory (e.g., factors, consequences). According to our research question, we associate the individual and the product properties to factors and the recommendation properties to the consequences.

To get a better grasp of the data, we analyze the relation among individual properties, product properties and recommendation properties. Subsequently, we analyze the cold-start behavior of HCF. More precisely, what properties have what effect on the recommendation performance. For this purpose, we compare pairwise individual properties and product properties and their effect on the recommendation performance, which we have defined in Section 7.1.2.

In the following, we present the analysis of the relation among individual and product properties followed by the analysis of the effect of these properties on the recommendation performance.

Relation among Individual and Product Properties

We computed Pearson's correlation among individual and product properties for the MovieLens 100k dataset. The Pearson's correlations are pre-

Property	Effort	Attitude	Selectivity	Visibility	Popularity	Polarization
Effort	1	-0.209	0.026	-0.211	-0.075	-0.037
Attitude	-0.209	1	-0.305	0.090	0.162	-0.085
Selectivity	0.026	-0.305	1	-0.037	-0.081	0.060
Visibility	-0.211	0.090	-0.037	1	0.506	-0.214
Popularity	-0.075	0.162	-0.081	0.506	1	-0.550
Polarization	-0.037	-0.085	0.060	-0.214	-0.550	1

Table 7.1: Pearson's correlations between individual and product properties for the MovieLens 100k dataset. All values are statistically significant on the significance level $\alpha = 0.01$.

sented in Table 7.1. All values are statistically significant on the significance level $\alpha = 0.01$.

The individuals' effort correlates negatively and weakly (-0.209) with the individuals' attitude. This indicates that individuals which provide few ratings tend to consume popular products. This is indicated by the positive and weak correlation (0.162) of individuals' attitude and products' popularity.

Furthermore, the effort correlates negatively and weakly (-0.211) with visibility. In other words, individuals which consume many products tend to expand their search for interesting products to the long-tail of products.

The individuals's attitude correlates negatively and moderately (-0.305) with individuals' selectivity. This indicates that individuals with high attitude tend to focus on popular products (0.162) which provide high satisfaction in general. In turn, individuals with low attitude are picky.

The products' visibility correlates positively and strongly (0.506) with the products' popularity. This means that popular products gets consumed most.

Furthermore, the products visibility correlates negatively and weakly (-0.214) with products' polarization. This may be because individuals get uncertain based on individuals' divergent opinions about polarizing products. In the case of popular products, individuals generally agree that these products provide high satisfaction. This is indicated by the negatively and strongly correlation (-0.550) of products' popularity and products' polarization.

Effect of Individual Properties and Product Properties to Recommendation Performance

The pairwise effect of individual properties and product properties on the recommendation performance are plotted on a 3-dimensional coordination system. The recommendation performance in terms of MAE is presented on the z-axis. A color scales supports the assessment of the MAE's magnitude. Red colored fields and high MAE, respectively, indicate low recommendation performance. In contrast, blue colored fields or low MAE, respectively, indicate a high recommendation performance. As discussed in Section 7.1.2, we bin the individuals and products to 20-quantiles regarding to their properties.

In the following, we analyze the effect of individual and product properties to the recommendation performance of HCF by means of the best performing method *HCkNN-NBSup*[1].

Effort and Visibility. The effect of individuals' effort and products' visibility is presented in Figure 7.1. Generally, the less visible a product is, the lower the recommendation performance. The reason is that low visible products, by definition, are rated by few individuals. Generally in collaborative filtering, these individuals are exclusively considered to predict the rating for a particular individual. However, the chance to retrieve like-minded individuals from few individuals is small, thus collaborative filtering providing poor recommendations.

In contrast, the individuals' effort has a marginal effect on recommendation performance, in particular individuals with low effort. This indicates

[1]See Section 6.4 for the discussion of the evaluation results.

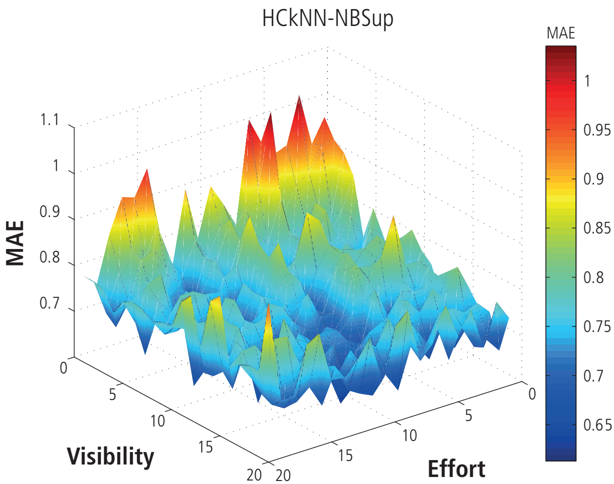


Figure 7.1: Distribution of recommendation performance with respect to effect and visibility.

the strength of retrieving like-minded individuals based on the comparison of the similarity of hypothesized preferences. For comparison, individuals' effort has a substantial effect on the recommendation performance when retrieving like-minded individuals based on the comparison of the rating similarity for common rated products (see Figure D.1e in Appendix D). This is referred to as the new-user cold-start problem as described in Section 2.2.2.

Effort and Popularity. The effect of individuals' effort and products' popularity is presented in Figure 7.2. Generally, the more popular products are, the better the recommendation performance. The reason is trivial. Products which satisfy most individuals become popular, thus being commonly a good recommendation. The accuracy of retrieving like-minded individuals which have rated these products is of less importance.

On the other hand, the less popular products are, the poorer the recom-

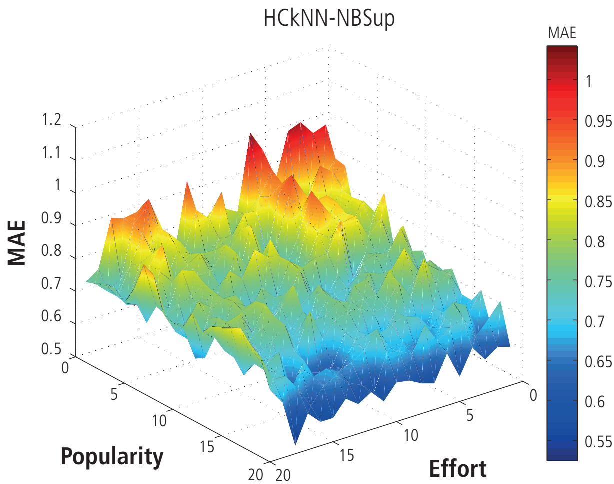


Figure 7.2: Distribution of recommendation performance with respect to effort and popularity.

mendation performance. This is counterintuitive since unpopular products typically provide low satisfaction to individuals. Referring to Table 7.1, less popular products are generally less visible. Therefore, the set of potential like-minded individuals is small and hence reducing the chance to find like-minded individuals. As a consequence, collaborative filtering provides provides poor recommendations.

Effort and Polarization The effect of individuals' effort and products' polarization is presented in Figure 7.3. Generally, the less products polarize, the better the recommendation performance. The reason is trivial. The majority of individuals agree to a certain rating for a product, which commonly holds for other individuals. Therefore, the retrieval of like-minded individuals is of less importance.

In turn, the more products polarize, the poorer the recommendation per-

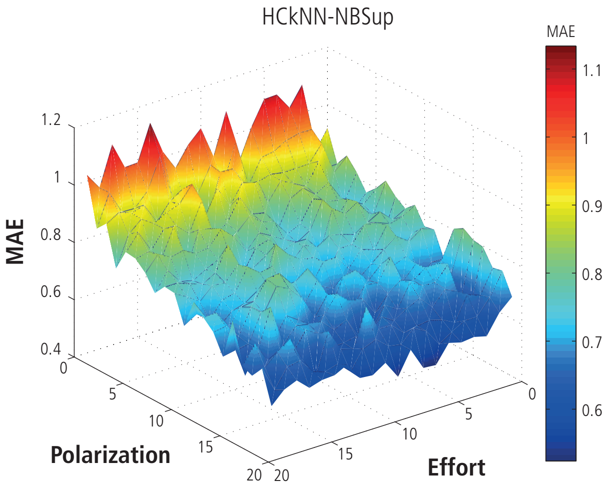


Figure 7.3: Distribution of recommendation performance with respect to effort and polarization.

formance. For these products, considering an individual's preferences gets crucial to provide personalized recommendations. Collaborative filtering aims to improving recommendation performance for exactly these kind of products. For comparison, Figure D.3f in Appendix D shows the substantial drop in recommendation performance of WoC the higher individuals' polarization gets.

Generally, the individual's effort has a less effect on the recommendation performance compared to the individual's polarization. In other words, independent from the amount of information a recommender system has, the utility that a polarizing products provides to individuals is difficult to predict.

Attitude and Visibility. The effect of individuals' attitude and products' visibility is presented in Figure 7.4. Generally, the higher an individual's

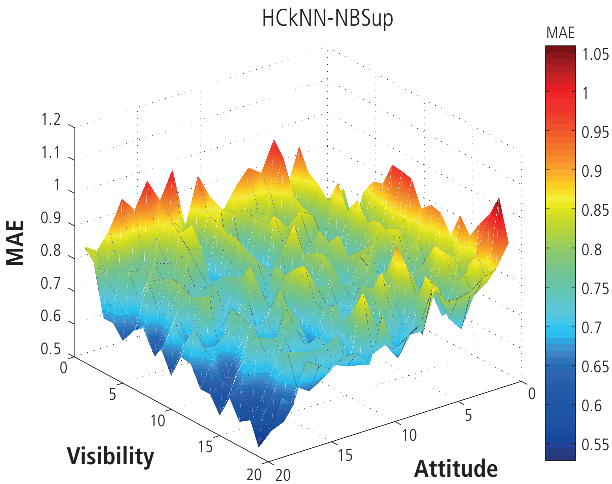


Figure 7.4: Distribution of recommendation performance with respect to attitude and visibility.

attitude, the better the recommendation performance. The reason is that most products provide high utility to these individuals. Furthermore, they are less selective as it is indicated by Table 7.1. In contrast, the lower an individual's attitude, the lower the recommendation performance. The reason is that they are more picky as it is indicated by Table 7.1. Nonetheless, collaborative filtering methods provide better recommendations to picky individuals than other methods as it is confirmed by Figure D.4f in Appendix D.

Apart from the least visible products, the product's visibility has a less effect on the recommendation performance. The products with least visibility, however, provide only a small set of potential like-minded individuals what limits the chance of finding like-minded individuals. Hence, ratings of any individuals are considered, thus collaborative filtering providing poor recommendations.

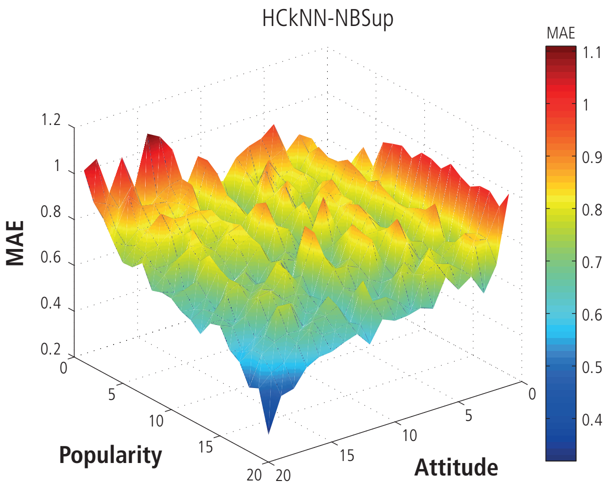


Figure 7.5: Distribution of recommendation performance with respect to attitude and popularity to MAE.

Attitude and Popularity. The effect of individuals' attitude and products' popularity is presented in Figure 7.5. The effect of attitude and popularity behaves similar to the effect of attitude and visibility. This is due to the strong correlation between popularity and visibility as it is shown in Table 7.1.

As Figure 7.5 shows, recommending popular products to individuals with high attitude is trivial. Additionally, predicting that unpopular products provide low utility to individuals wit low attitude is likewise trivial as it is indicated in Figure 7.5.

Attitude and Polarization The effect of individuals' attitude and products' polarization is presented in Figure 7.6. Generally, the higher an individual's attitude and the lower a product's polarization, the better the recommendation performance. Referring to Table 7.1, the reason is that individuals

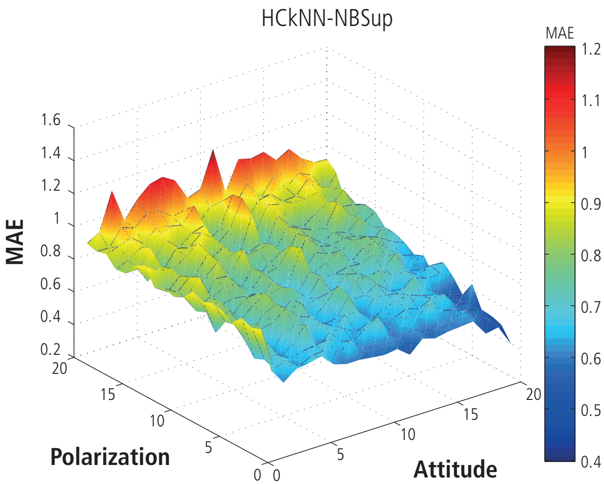


Figure 7.6: Distribution of recommendation performance with respect to attitude and polarization.

commonly agree on a rating for a popular products what reduces their polarization at the same time. Typically, popular products provide high utility to individuals, thus making the recommendation task trivial.

Furthermore, the recommendation performance for products with low polarization is high in general. This is due to the fact that products with low polarization typically provide similar utility to any individual what makes the recommendation task simple.

Selectivity and Visibility. The effect of individuals' selectivity and products' visibility is presented in Figure 7.7. Generally, the recommendation performance depends primarily on the individual's selectivity. The less selective an individual the better the recommendation performance. The reason is that in this particular case the utility of any product converges to the individual's attitude the less selective the individual is.

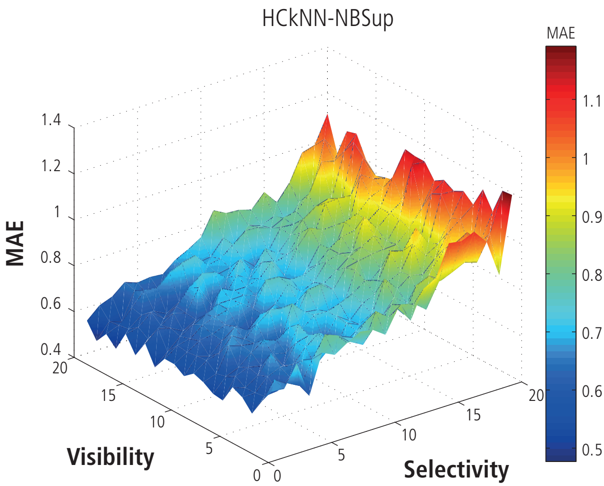


Figure 7.7: Distribution of recommendation performance with respect to selectivity and visibility.

On the other hand, the more selective an individual, the lower the recommendation performance.

Selectivity and Popularity. The effect of individuals' selectivity and products' popularity is presented in Figure 7.8. The distribution of the recommendation performance with respect to the selectivity and popularity is similar to the distribution with respect to the selectivity and visibility. Referring to Table 7.1, the reason is the strong correlation between visibility and popularity. In this case, however, popularity has some effect on the recommendation performance, although the effect is small. As it is shown in Figure 7.8, the higher a product's popularity the better the recommendation performance.

However, the higher the selectivity the lower the recommendation performance.

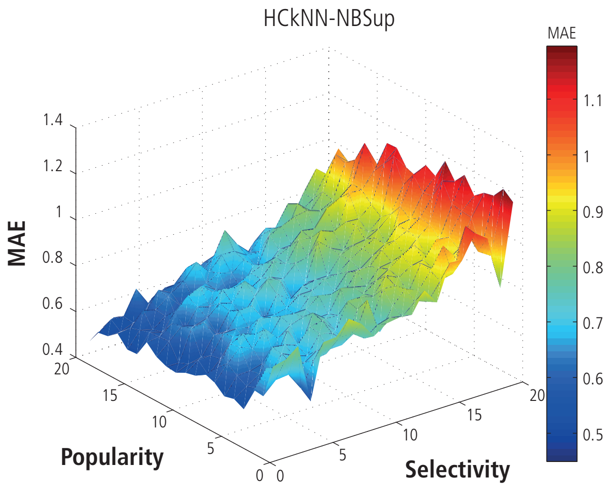


Figure 7.8: Distribution of recommendation performance with respect to selectivity and popularity.

Selectivity and Polarization The effect of individuals' selectivity and products' polarization is presented in Figure 7.9. Generally, the lower the selectivity and the lower the polarization, the better the recommendation performance. Products with low polarization are commonly popular products as it is indicated by the strong correlation of polarization and popularity in Table 7.1.

In contrast, the higher the selectivity and the higher the polarization, the lower the recommendation performance. High selective individuals make the recommendation tasks difficult. Polarizing products even enforce it. Nevertheless, the recommendation performance is high compared to the recommendation performance of WoC as it is shown in Figure D.9f in Appendix D.

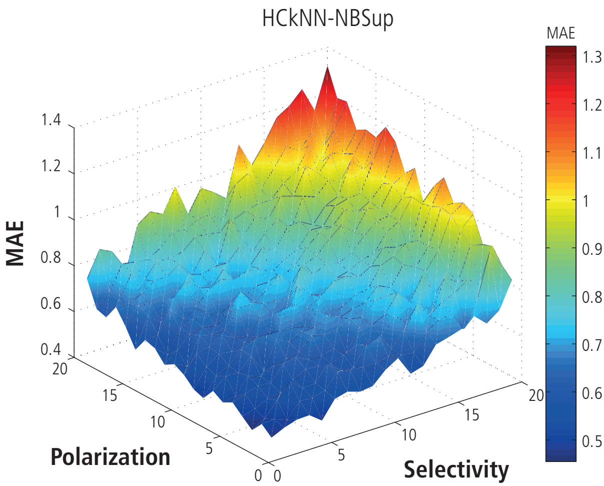


Figure 7.9: Distribution of recommendation performance with respect to selectivity and polarization.

7.2 Theory Development

In this section, we develop the grounded theory which explains the phenomenon of HCF performing better than other collaborative filtering methods. Above all, we investigate in what cases the retrieval of like-minded individuals based on the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products.

To this end, we compare three methods, which consider like-minded individuals to compute recommendations in terms of recommendation performance. The first method is *HCkNN-NBSup* (see Section 6.2.1), which retrieves like-minded individuals based on the comparison of the composition of hypothesized preferences as it is described in Section 5.3. As shown in Section 6.4, *HCkNN-NBSup* performs best among all other methods we considered for evaluation[2].

The second method is *HUkNN-NBcorr* (see Section 6.2.1), which retrieves like-minded individuals based on the comparison of hypothesized utilities for some products as it is described in Section 5.2. *HUkNN-NBcorr* performs best among other methods which retrieve like-minded individuals based on *hypothesized utility-based preferences similarity* (HU preference similarity).

The third method is the baseline collaborative filtering method PcorrkNN (see Section 6.2.2), which retrieves like-minded individuals based on the similarity of their ratings for common rated products. PcorrkNN performs best among the baseline methods for comparison as discussed in Section 6.4.

In the following, we first discuss the concepts to formulate the hypotheses which explain the phenomenon in Section 7.2.1. Then, we formulate the hypotheses which constitutes the grounded theory based on the comparison of the three methods in Section 7.2.2.

7.2.1 Theory Concepts

We obtain the concepts from the individual, product and recommendation properties, individuals' effort in particular, attitude and selectivity, products' visibility, popularity and polarization, and the recommendation properties ΔMAE PcorrkNN for the comparison of HCF and the baseline collaborative filtering method PcorrkNN. Furthermore, we use the the four issues introduced in Section 1.1, specifically similarity significance, partial preference representation, similarity assessability and preferences incompleteness.

The relative recommendation performance (ΔMAE PcorrkNN) of the HCF methods relative to PcorrkNN strongly correlates with individual and product properties as it is shown in Table 7.2 by means of the MovieLens 100k dataset. For this reason, we consider all individual properties (i.e., effort, attitude and selectivity) and all product properties (i.e., visibility, popularity, polarization) as concepts to formulate the hypotheses which

[2] The considered methods are introduced in Section 6.2

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.853	-0.546	0.781	-0.567	-0.236	-0.045
HUkNN-NBcorr	0.896	-0.657	0.472	-0.824	-0.606	0.191
HUkNN-SVMcorr	0.934	-0.315	0.242	-0.830	-0.560	0.044
HUkNN-NBcorr	0.895	-0.375	0.881	-0.587	-0.099	0.500
HUkNN-NBprob	0.912	-0.245	0.871	-0.704	0.035	0.240
HUkNN-sJ48prob	0.890	-0.619	0.795	-0.833	-0.672	0.403
HCkNN-J48NoG	0.884	-0.232	0.113	-0.937	-0.812	0.485
HCkNN-J48Sup	0.888	-0.220	0.196	-0.939	-0.807	0.451
HCkNN-NBSup	0.934	-0.179	0.537	-0.845	-0.397	0.118
HCkNN-J48SSup	0.915	-0.117	0.257	-0.886	-0.809	0.386
HCkNN-NBSSup	0.909	-0.493	0.840	-0.231	0.098	-0.682

Table 7.2: Pearson's correlations between recommendation performance difference and properties of individuals and products in the MovieLens 100k dataset. All values are statistically significant on the significance level $\alpha = 0.01$.

ultimately constitute the grounded theory. All Pearson's correlation values are statistically significant on the significance level $a = 0.01$.

Especially the Pearson's correlation of ΔMAE PcorrkNN and effort respectively visibility are strong for all methods. In case of the datasets with increasing sparsity, the Pearson's correlations remain significant and strong. We refer to Appendix E where the Pearson's correlation values for the remaining datasets with increasing sparsity are presented.

7.2.2 Comparison of Recommendation Performance

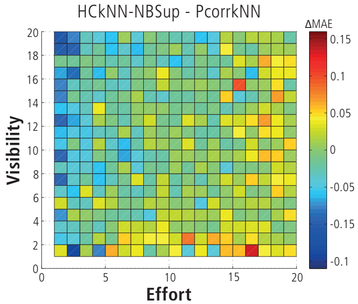
In the following, we compare the recommendation performance of both HCF methods, *HCkNN-NBSup* and *HUkNN-NBprob*, with the baseline collaborative filtering method PcorrkNN by means of individual and product properties. Based on the comparison, we formulate the hypotheses which constitutes the grounded theory.

We choose three datasets, which represent the major spectrum of sparsity, to observe the cold-start behavior and the distribution of the recommendation performance. These datasets are specifically the original MovieLens 100k dataset with 0% sparsity, which represents a minor cold-start problem, the dataset with 40% sparsity, which represents the transition from a minor cold-start problem to a major cold-start problem, and the dataset with 80% sparsity, which represents a major cold-start problem. The relative recommendation performance is measured by means of ΔMAE PcorrkNN and is encoded on a color scale. Blue colored fields indicate the superiority of HCF to PcorrkNN. In contrast, red colored fields indicate the inferiority of HCF to PcorrkNN.

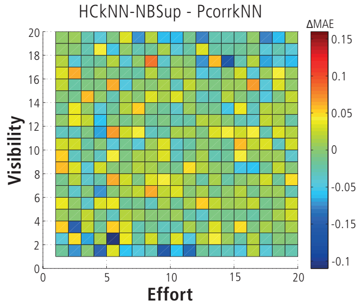
Effort and Visibility

Figure 7.10 and Figure 7.11 present the comparison of recommendation performance by means of effort and visibility. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

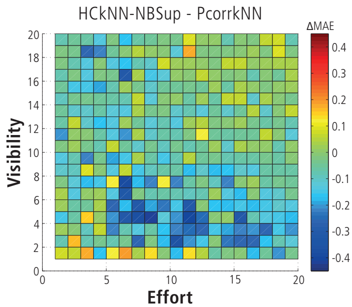
Low effort hypothesis. As shown in Figure 7.10a and Figure 7.11a, the lower the effort the better the recommendation performance. The reason is that individuals with low effort provide by definition few ratings, which limits substantially the set of common rated products. This adds to the issue of similarity significance (see Section 1.1) because the preference similarity is computed based on a small set of common rated products and therefore may not be significant. Additionally, the common rated products with respect to other individuals may represent only partially their preferences, thus reinforcing the issue of partial preference representation (see Section 1.1). Furthermore, due to the small set of common rated products, the chance is high that like-minded individuals have no common rated products, thus not being retrieved. This refers to the issue of similarity assessability (see Section 1.1).



(a) Dataset with with 0% sparsity.

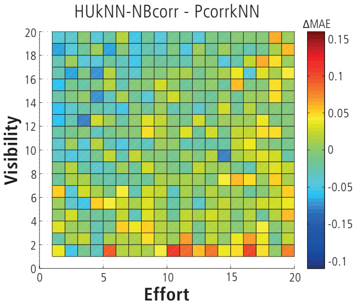


(b) Dataset with 40% sparsity.

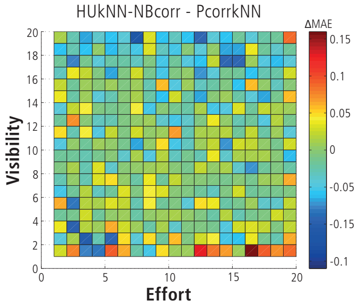


(c) Dataset with 80% sparsity.

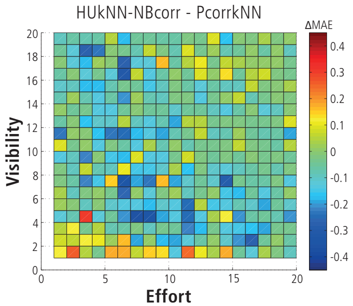
Figure 7.10: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' effort and products' visibility.



(a) Dataset with 0% sparsity.



(b) Dataset with 40% sparsity.



(c) Dataset with 80% sparsity.

Figure 7.11: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' effort and products' visibility.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH1): *Individuals with low effort add to the issues of similarity significance, partial preference representation and similarity assessability. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

High effort hypothesis. As indicated in Figure 7.10a and Figure 7.11a, both *HCkNN-NBSup* and *HUkNN-NBcorr* are inferior to PcorrkNN in the case of individuals with high effort. As we discussed in Section 6.5, the more ratings an individual providse the better these ratings represent the individual's preferences. In contrast, the accuracy of hypothesizing preferences is limited due to the limited hypothesis space. The information gain of additional ratings converges to zero, thus limiting the accuracy of hypothesized preferences.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH2): *The more ratings an individual provides the better these ratings represent the individual's preferences. In contrast, the limited hypothesis space limits the accuracy of hypothesized preferences. Hence, the comparison of hypothesized preferences is inferior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

High visibility hypothesis. In the case of products with high visibility the set of potential like-minded individuals is large. When comparing ratings for common rated products, many individuals are wrongly retrieved as like-minded due to partial preference similarity. In other words, products' high visibility adds to the issue of partial preference representation. As indicated by Figure 7.10a and Figure 7.11a, retrieving like-minded individuals based on the comparison of hypothesized preferences instead of the comparison of ratings for common rated products mitigates these issues and consequently both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH3): *Products with high visibility provide a large set of potential like-minded individuals. Due to the issue of partial preference representation, many individuals are wrongly retrieved as like-minded. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Low effort - high visibility hypothesis. Especially in the case of minor cold-start problem (see Figure 7.10a and Figure 7.11a), both *HCkNN-NBSup* and *HUkNN-NBcorr* perform better compared to PcorrkNN the lower the individual's effort and the higher the product's visibility. The reasons are twofold. Firstly, individuals with low effort provide by definition few ratings, which limits substantially the set of common rated products. This adds to the issue of similarity significance because the preference similarity is computed based on a small set of common rated products and therefore may not be significant. Additionally, the common rated products with respect to other individuals may represent only partially their preferences, thus reinforcing the issue of partial preference representation. Furthermore, due to the small set of common rated products, the chance is high that like-minded individuals have no common rated products, thus not being retrieved. This refers to the issue of similarity assessability. This hypothesis combines both Hypotheses GH1 and GH3.

Secondly, in the case of products with high visibility the set of potential like-minded individuals is large. Combined with individuals with low effort, many individuals are wrongly retrieved as like-minded due to partial preference similarity. In other words, the issue of partial preference representation is reinforced.

As indicated by Figure 7.10a and Figure 7.11a, retrieving like-minded individuals based on the comparison of hypothesized preferences instead of the comparison of ratings for common rated products mitigates these issues and consequently *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH4): *Individuals with low effort and products with high visibility add to the issues of similarity significance, partial preference representation and similarity assessability. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Note that this Hypothesis GH4 follows from the Hypotheses GH1 and GH3.

Low visibility in minor cold-start situation hypothesis. In the case of minor cold-start situation, both *HCkNN-NBSup* and *HUkNN-NBcorr* are inferior to PcorrkNN regarding the recommendation performance for products with low visibility. The reason is that the comparison of hypothesized preferences provides a better ranking of like-mindedness than the comparison of ratings for common rated products. We base this argument on the analysis of similarity values computed by HCF methods. Products with low visibility provide a small set of individuals for collaborative filtering. Thus, weighting the like-mindedness instead of ranking gets more crucial. Referring to Figure 7.10c and Figure 7.11c. In the case of major cold-start problem, the issue of similarity assessability outweights this issue as indicated , however.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH5): *Comparison of hypothesized preferences provides better ranking of like-mindedness. In contrast, the comparison of ratings for common rated products provides a better weighting of like-mindedness. Hence, the* *comparison of hypothesized preferences is inferior to the comparison of ratings for common rated products to retrieve like-minded individuals in the case of products with low visibility in minor cold-start situations.*

Low visibility in major cold-start situation hypothesis. In the case of major cold-start problem (see Figure 7.10c and Figure 7.11c), both *HCkNN-NBSup* and *HUkNN-NBcorr* substantially outperform PcorrkNN in the case of products with low visibility. The reason is that products with low visibility provide by definition a small set of potential like-minded individuals, which limits the chance of finding like-minded individuals who have rated these products. Since individuals provide few ratings in general in major cold-start situations, like-minded individuals are missed. Therefore, products with low visibility in major cold-start situations add to the issue of similarity assessability. In contrast, HCF is superior because it allows for the comparison of individuals' preferences based on their hypothesized preferences, which is independent from common rated products.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH6): *In a major cold-start situation, products with low visibility reinforce the issue of similarity assessability. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Low effort - low visibility in major cold-start situation hypothesis. Figure 7.11c shows that *HUkNN-NBcorr* is inferior to PcorrkNN in the case of individuals with low effort and products with low visibility. The basic reason is that individuals provide insufficient ratings to adequately hypothesize their preferences with machine learning. Furthermore, the advantage of using the set of unified rated products, which is larger than the set common rated products, is too small to compensate for inaccurate hypothesized preferences. Products with low visibility limits the set of potential like-minded individuals, thus making the retrieval of like-minded individuals more error-prone. However, the higher the individuals' effort the more accurate their respective hypothesized preferences. As a result, the relative recommendation performance increases. Nevertheless, both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN for major cold-start situations in general.

In conclusion, we formulate the following hypothesis:

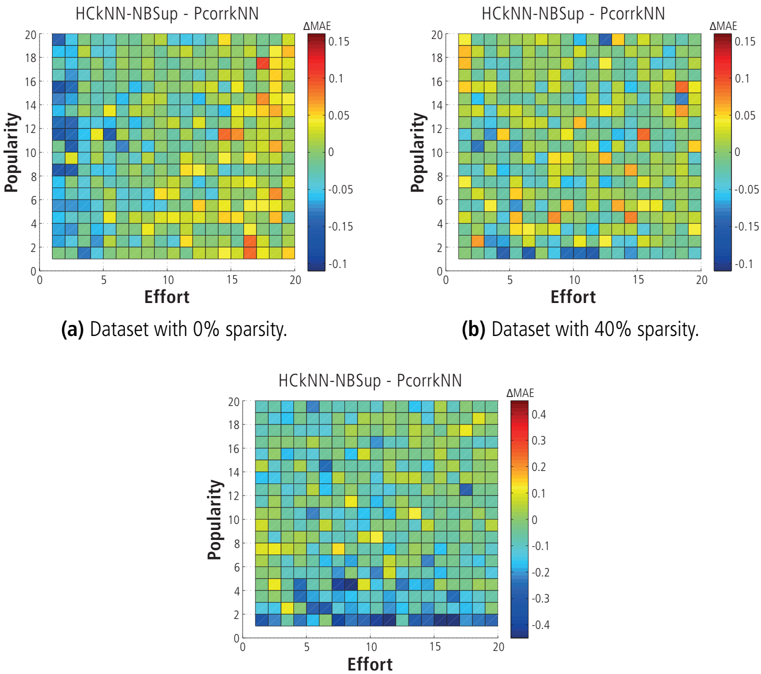
GROUNDED HYPOTHESIS (GH7): *In a major cold-start situation, hypothesized preferences of individuals with low effort are less accurate. In the case of HU collaborative filtering methods, the advantage of using the larger set of unified rated products does not compensate for inaccurate hypothesized preferences. Hence, the comparison of hypothesized preferences by means of hypothesized utilities for some products is inferior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Effort and Popularity

Figure 7.12 and Figure 7.13 present the comparison of recommendation performance by means of effort and popularity. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

Both *HCkNN-NBSup* and *HUkNN-NBcorr* perform relative better the lower the individual's effort and the higher the product's popularity in the minor cold-start situation as it is shown in Figure 7.12a and Figure 7.13a. The product's popularity, however, influences less the relative recommendation performance of *HCkNN-NBSup*. Hence, we confirm Hypothesis GH1 (low effort hypothesis).

Furthermore, PcorrkNN outperforms *HCkNN-NBSup* and *HUkNN-NBcorr* in the case of individuals with high effort. Hence, we confirm Hypothesis GH2 (high effort hypothesis).

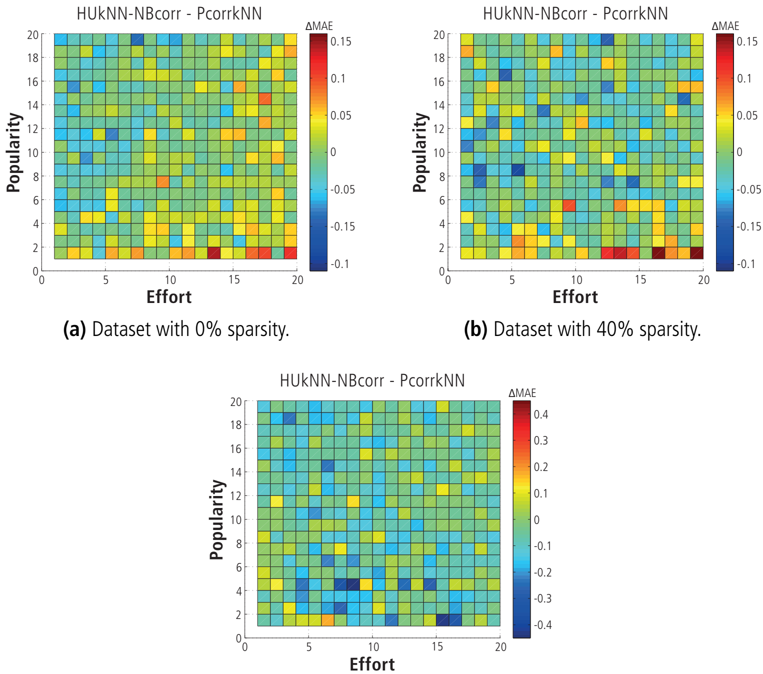


(a) Dataset with 0% sparsity.

(b) Dataset with 40% sparsity.

(c) Dataset with 80% sparsity.

Figure 7.12: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' effort and products' popularity.



(a) Dataset with 0% sparsity.

(b) Dataset with 40% sparsity.

(c) Dataset with 80% sparsity.

Figure 7.13: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' effort and products' popularity.

Low popularity in major cold-start situation hypothesis. Referring to Figure 7.12c and Figure 7.13c, both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN the major the cold-start problem gets, especially in the case of unpopular products. This confirms the evaluation results of the relevance filtering quality in Section 6.4.2, which show the superiority of *HCkNN-NBSup* and *HUkNN-NBcorr* to PcorrkNN regarding relevance filtering quality.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH8): *HCF better filters unpopular products in major cold-start situations independent of the individual's effort. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

High effort - low popularity in minor cold-start situation hypothesis. Referring to Figure 7.12a and Figure 7.13a, both *HCkNN-NBSup* and *HUkNN-NBcorr* are inferior to PcorrkNN in the minor cold-start situation regarding individuals with high effort and unpopular products. The reasons are twofold. Firstly, the more ratings individuals provide the better these ratings represent the individuals' preferences. In contrast, the accuracy of hypothesized preferences is limited due to the limited hypothesis space. The information gain of additional ratings converges to zero, thus limiting the accuracy of hypothesized preferences.

Secondly, popularity strongly correlates with visibility (see Table 7.1 in Section 7.1.3). Hence, the same reason holds as for visibility (see explanation in Hypothesis GH5).

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH9): *The Comparison of hypothesized preferences provides better ranking of like-mindedness. In contrast, the comparison of ratings for common rated products provides a better weighting of like-mindedness. The* *more ratings individuals provide the better these ratings represent the individuals' preferences. In contrast, the limited hypothesis space limits the accuracy of hypothesized preferences. Hence, the comparison of hypothesized preferences is inferior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Low effort - high popularity in minor cold-start situation hypothesis. Referring to Figure 7.12a and Figure 7.13a, both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN in the minor cold-start situation regarding individuals with low effort and popular products. The reasons are twofold. Firstly, individuals with low effort provide by definition few ratings, which limits substantially the set of common rated products. This adds to the issue of similarity significance because the preference similarity is computed based on a small set of common rated products and therefore may not be significant. Additionally, the common rated products with respect to other individuals may represent only partially their preferences, thus reinforcing the issue of partial preference representation. Furthermore, due to the small set of common rated products, the chance is high that like-minded individuals have no common rated products, thus not being retrieved. This refers to the issue of similarity assessability.

Secondly, popularity strongly correlates with visibility (see Table 7.1 in Section 7.1.3). Hence, the same reason holds for products with high popularity as for products with high visibility (see explanation in Hypothesis GH3).

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH10): *Individuals with low effort and products with high popularity add to the issues of similarity significance, partial preference representation and similarity assessability. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Effort and Polarization

Figure 7.14 and Figure 7.15 present the comparison of recommendation performance by means of effort and polarization. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem. In the case of major cold-start problem, both *HCkNN-NBSup* and *HUkNN-NBprob* are superior to PcorrkNN in general. As Figure 7.14c and Figure 7.15c show, the relative recommendation performance is mainly uniformly distributed.

Referring to Figure 7.14a and Figure 7.15a, the lower the individual's effort the superior are *HCkNN-NBSup* and *HUkNN-NBprob* to PcorrkNN. Hence, we confirm Hypothesis GH1 (low effort hypothesis).

In turn, the higher the individual's effort the more inferior *HCkNN-NBSup* and *HUkNN-NBprob* to PcorrkNN. Hence, we confirm Hypothesis GH2 (high effort hypothesis).

Polarization independence hypothesis. As shown in Figure 7.15, the polarization has a marginal, if not none, effect on the relative recommendation performance.

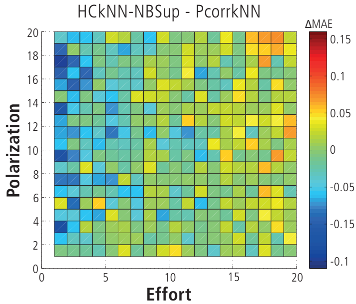
Therefore, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH11): *The relative recommendation performance is independent from the product's polarization.*

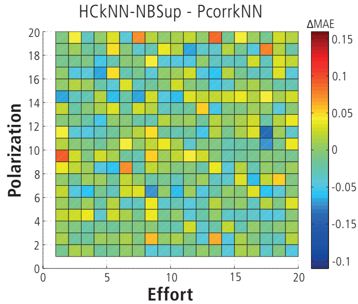
Attitude and Visibility

Figure 7.16 and Figure 7.17 present the comparison of recommendation performance by means of attitude and visibility. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

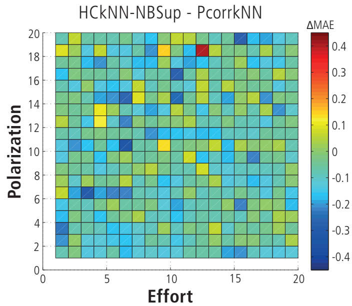
As shown in Figure 7.16a and Figure 7.17a, the higher the visibility the more superior are *HCkNN-NBSup* and *HUkNN-NBcorr* to PcorrkNN. On the



(a) Dataset with 0% sparsity.

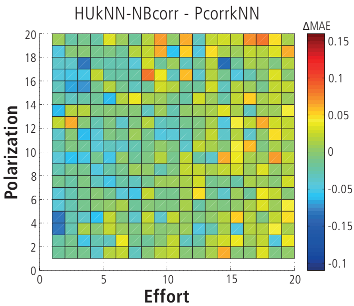


(b) Dataset with 40% sparsity.

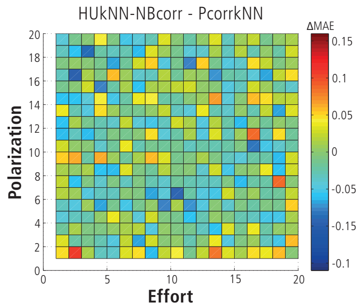


(c) Dataset with 80% sparsity.

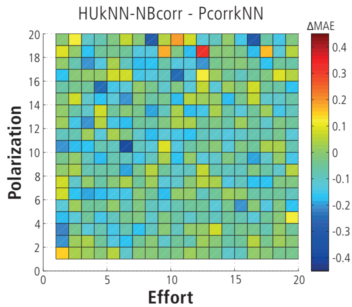
Figure 7.14: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' effort and products' polarization.



(a) Dataset with 0% sparsity.

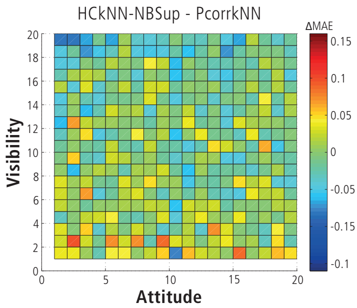


(b) Dataset with 40% sparsity.

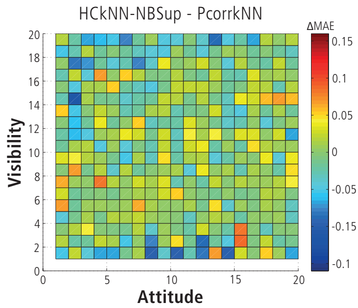


(c) Dataset with 80% sparsity.

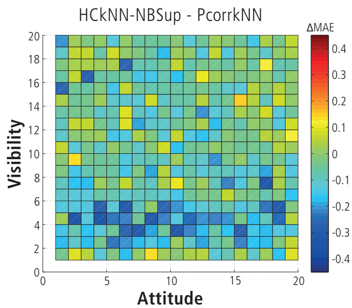
Figure 7.15: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' effort and products' polarization.



(a) Dataset with 0% sparsity.

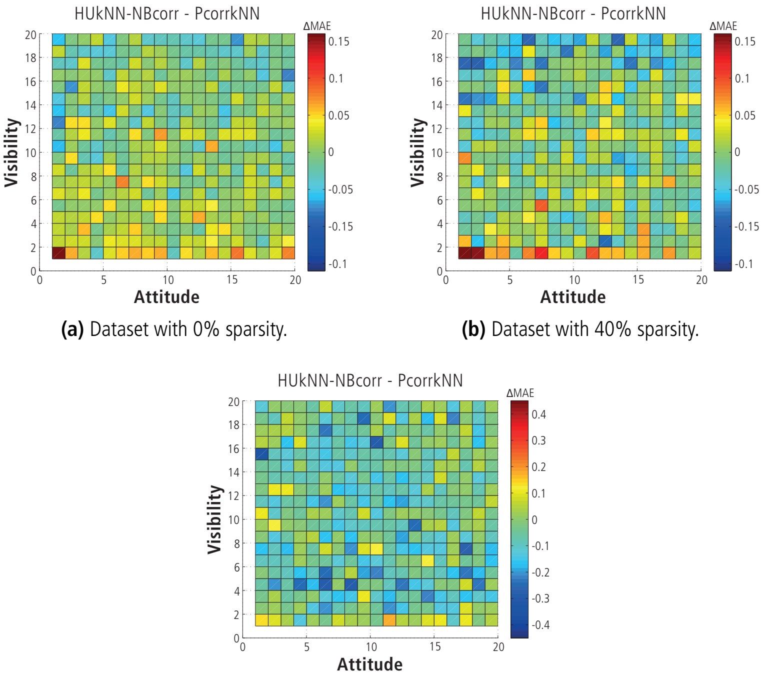


(b) Dataset with 40% sparsity.



(c) Dataset with 80% sparsity.

Figure 7.16: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' attitude and products' visibility.



(a) Dataset with 0% sparsity.

(b) Dataset with 40% sparsity.

(c) Dataset with 80% sparsity.

Figure 7.17: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' attitude and products' visibility.

other hand, the lower the visibility the more inferior are *HCkNN-NBSup* and *HUkNN-NBcorr* to PcorrkNN. Hence, we confirm Hypothesis GH3 (high visibility hypothesis) and Hypothesis GH5 (low visibility in minor cold start situation hypothesis).

In the case of major cold-start situation, *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN in general, especially for products with low visibility. This is shown in Figure 7.16c and Figure 7.17c. Hence, we confirm Hypothesis GH6 (low visibility in major cold-start situation hypothesis).

Like-minded individuals retrieval precision hypothesis. Referring to Figure 7.16a and Figure 7.17a, the higher the product's visibility, the better the relative recommendation performance of *HCkNN-NBSup* and *HUkNN-NBcorr*. This holds also for the transition phase as Figure 7.16b and Figure 7.17b confirm. This means that retrieving like-minded individuals based on the comparison of hypothesized preferences is more precise than based on the comparison of ratings for common rated products.

In conclusion, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH12): *The comparison of hypothesized preferences is more precise than the comparison of ratings for common rated products, specifically when the set of potential like-minded individuals is large. Therefore, retrieving like-minded individuals based on the comparison of hypothesized preferences should be favored.*

Attitude independence hypothesis. The individual's attitude has no effect on the relative recommendation performance as it is confirmed by Figure 7.16 and Figure 7.17.

Therefore, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH13): *The relative recommendation performance is independent from the individual's attitude.*

Attitude and Popularity

Figure 7.18 and Figure 7.19 present the comparison of recommendation performance by means of attitude and popularity. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

Both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN the more major the cold-start problem gets, especially in the case of unpopular products. This is shown in Figure 7.18c and Figure 7.19c. Hence, we confirm Hypothesis GH8 (low popularity in major cold-start situation hypothesis).

The individual's attitude has no effect on the relative recommendation performance as it is confirmed by Figure 7.18 and Figure 7.19. Hence, we confirm Hypothesis GH13 (attitude independence hypothesis).

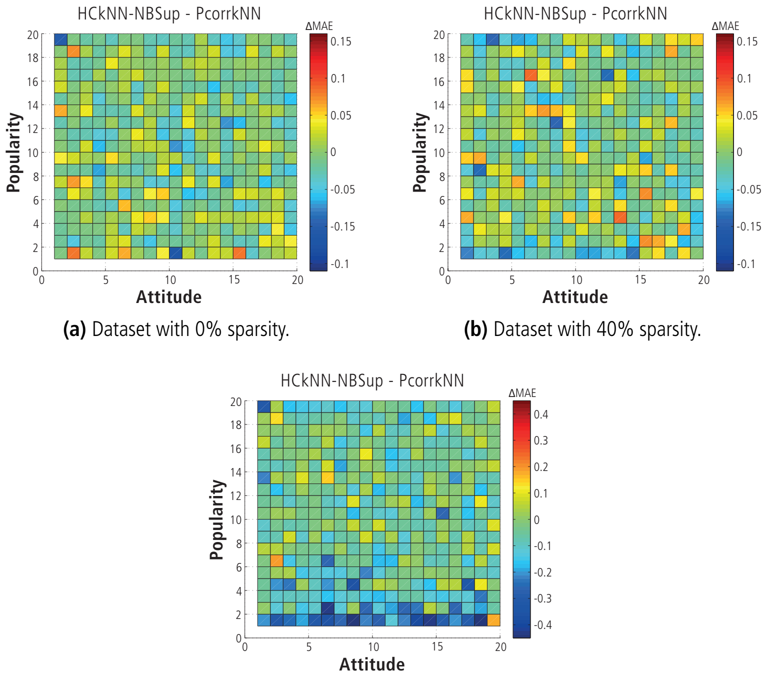
Attitude and Polarization

Figure 7.20 and Figure 7.21 present the comparison of recommendation performance by means of attitude and polarization. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

The individual's attitude as well as the product's polarization have no effect on the relative recommendation performance as it is confirmed by Figure 7.18 and Figure 7.19. Hence, we confirm Hypothesis GH11 (polarization independence hypothesis) and Hypothesis GH13 (attitude independence hypothesis).

Selectivity and Visibility

Figure 7.22 and Figure 7.23 present the comparison of recommendation performance by means of selectivity and visibility. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

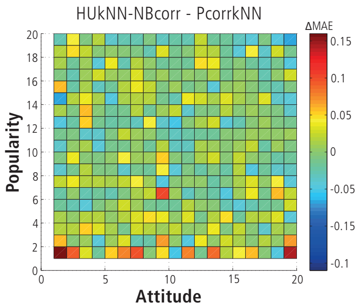


(a) Dataset with 0% sparsity.

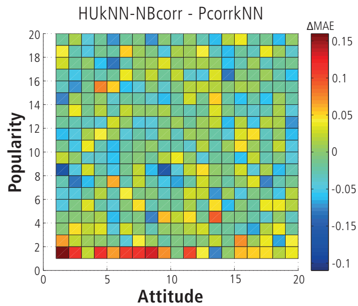
(b) Dataset with 40% sparsity.

(c) Dataset with 80% sparsity.

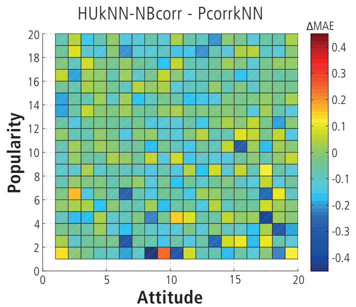
Figure 7.18: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' attitude and products' popularity.



(a) Dataset with 0% sparsity.

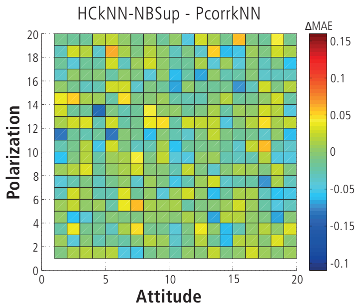


(b) Dataset with 40% sparsity.

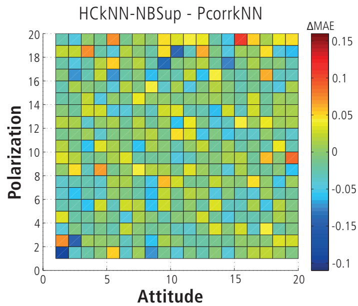


(c) Dataset with 80% sparsity.

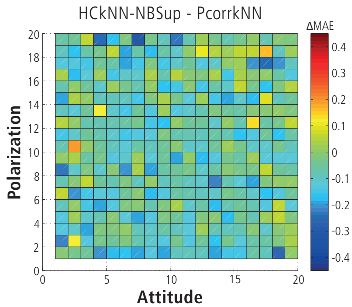
Figure 7.19: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' attitude and products' popularity.



(a) Dataset with 0% sparsity.

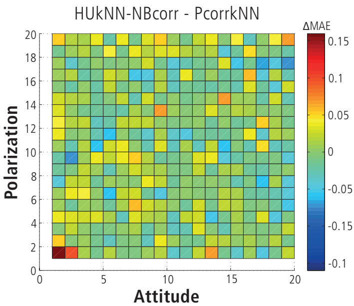


(b) Dataset with 40% sparsity.

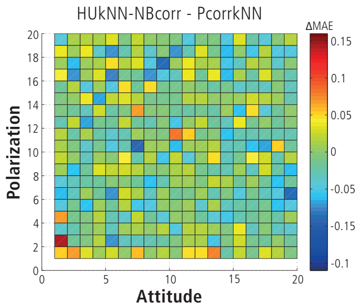


(c) Dataset with 80% sparsity.

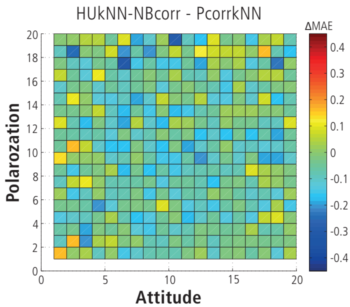
Figure 7.20: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' attitude and products' polarization.



(a) Dataset with 0% sparsity.

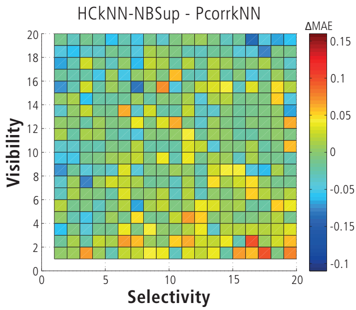


(b) Dataset with 40% sparsity.

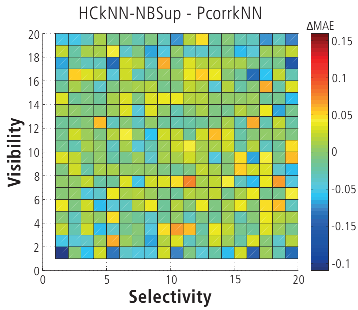


(c) Dataset with 80% sparsity.

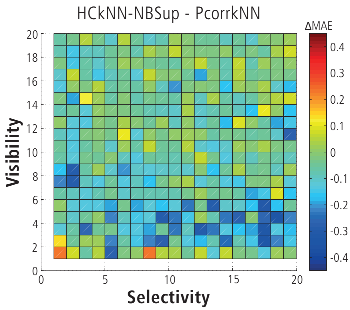
Figure 7.21: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' attitude and products' polarization.



(a) Dataset with 0% sparsity.



(b) Dataset with 40% sparsity.



(c) Dataset with 80% sparsity.

Figure 7.22: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' selectivity and products' visibility.

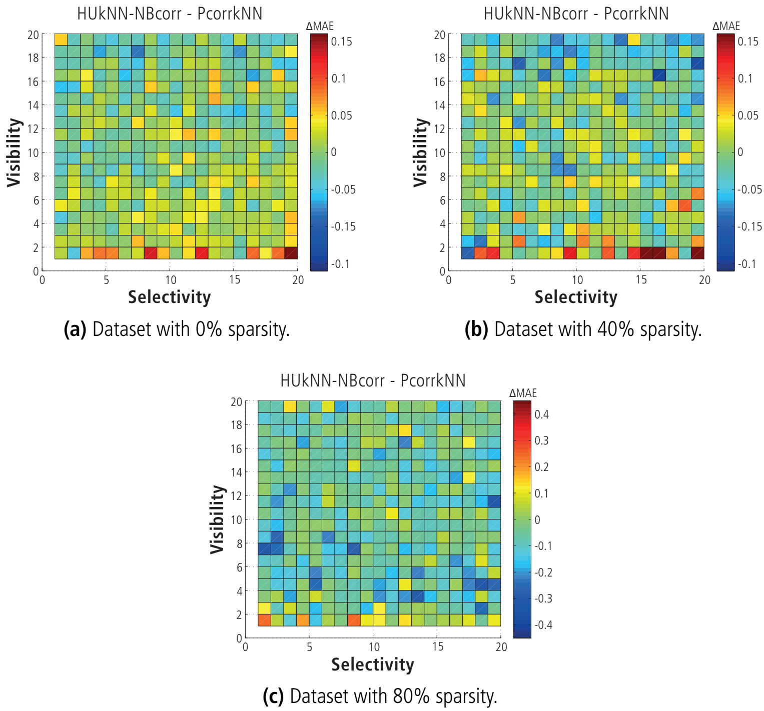


Figure 7.23: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' selectivity and products' visibility.

Referring to Figure 7.22a and Figure 7.23a, the higher the product's visibility the more superior are *HCkNN-NBSup* and *HUkNN-NBcorr* to PcorrkNN. In contrast, the lower the product's visibility the more inferior are *HCkNN-NBSup* and *HUkNN-NBcorr* to PcorrkNN. Hence, we confirm Hypothesis GH3 (high visibility hypothesis) and Hypothesis GH5 (low visibility in minor cold-start situation hypothesis).

In the case of major cold-start problem, both *HCkNN-NBSup* and *HUkNN-NBcorr* outperform PcorrkNN, particularly when the product's visibility is low. This is due to the small set of potential like-minded individuals which adds to the issue of similarity assessability. Hence, we confirm Hypothesis GH6 (low visibility in major cold-start situation hypothesis)

Low selectivity hypothesis. The relative recommendation performance between *HUkNN-NBcorr* and PcorrkNN regarding the individual's selectivity is small, however. This refers to the general issue of correlation-based similarity metrics which are better suited when the individual's selectivity is high. In conclusion, individuals with low selectivity add to the issue of similarity significance.

Therefore, we formulate the following hypothesis:

GROUNDED HYPOTHESIS (GH14): *Individuals with low selectivity add to to the issue of similarity significance. Especially correlation-based similarity metrics are inappropriate to compare the preferences of individuals with low selectivity to other individuals' preferences.*

Low selectivity - high visibility hypothesis. Referring to both figures Figure 7.22a and Figure 7.23a, both *HCkNN-NBSup* and *HUkNN-NBcorr* perform relative better the lower the individual's selectivity and the higher the product's visibility. The reasons are twofold. Firstly, individuals with low selectivity receive similar utility from any product. Typically, correlation-based preference similarity metrics are inappropriate to compare the preferences of individuals with low selectivity to others. In other words, individuals with low selectivity add to the issue of similarity significance.

Secondly, in the case of products with high visibility the set of potential like-minded individuals is large. Combined with individuals with low selectivity, many individuals are wrongly retrieved as like-minded due to partial preference similarity. This adds to the issue of partial preference representation.

In conclusion, we formulate the following hypothesis:

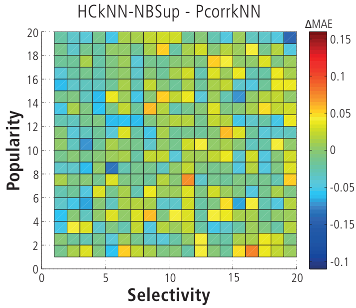
GROUNDED HYPOTHESIS (GH15): *Individuals with low selectivity and products with high visibility add to the issues of similarity significance, partial preference representation and similarity assessability. Hence, the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products to retrieve like-minded individuals.*

Selectivity and Popularity

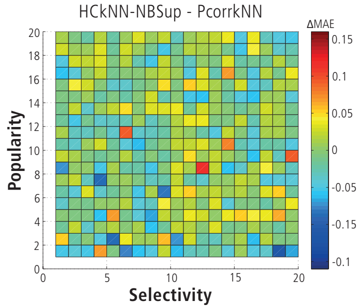
Figure 7.24 and Figure 7.25 present the comparison of recommendation performance by means of selectivity and popularity. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

Both *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN the more major the cold-start problem gets, especially when the product is unpopular. This is shown in Figure 7.24c and Figure 7.25c. Hence, we confirm Hypothesis GH8 (low popularity in major cold-start situation hypothesis).

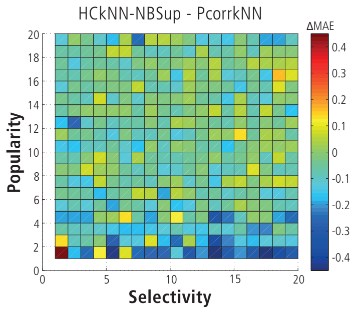
Referring to Figure 7.22a and Figure 7.23a, *HCkNN-NBSup* performs relative better the lower the individual's selectivity. The selectivity has marginal effect on the relative recommendation performance of *HUkNN-NBcorr*, however. Hence, we confirm Hypothesis GH14 (low selectivity hypothesis).



(a) Dataset with 0% sparsity.

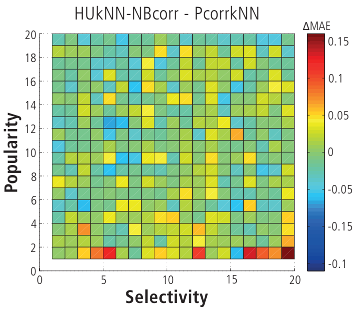


(b) Dataset with 40% sparsity.

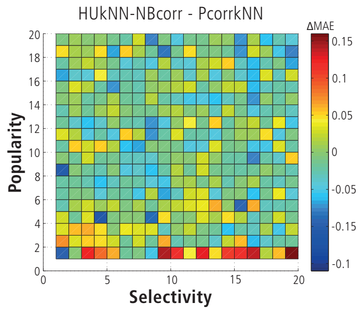


(c) Dataset with 80% sparsity.

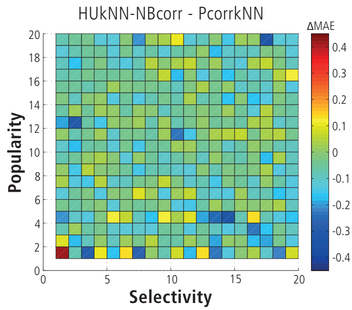
Figure 7.24: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' selectivity and products' popularity.



(a) Dataset with 0% sparsity.



(b) Dataset with 40% sparsity.



(c) Dataset with 80% sparsity.

Figure 7.25: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' selectivity and products' popularity.

Selectivity and Polarization

Figure 7.26 and Figure 7.27 present the comparison of recommendation performance by means of selectivity and polarization. Generally, the relative recommendation performance improves with increasing degree of the cold-start problem.

The individual's polarization has no effect on the relative recommendation performance as it is confirmed by Figure 7.18 and Figure 7.19. Hence, we confirm Hypothesis GH11 (polarization independence hypothesis).

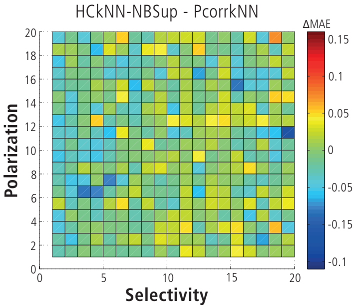
Figure 7.26a and Figure 7.27a show that *HCkNN-NBSup* and *HUkNN-NBcorr* are superior to PcorrkNN the lower the individual's selectivity. Hence, we confirm Hypothesis GH14 (low selectivity hypothesis).

7.3 Theory Consolidation

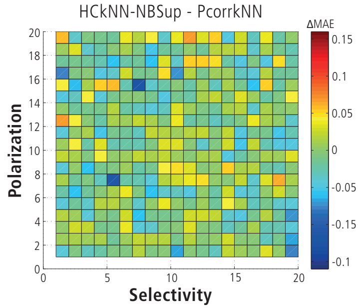
In the following, we consolidate the grounded hypotheses which we formulated in the previous section to constitute the grounded theory. Figure 7.28 presents the grounded theory showing when the comparison of hypothesized preferences is superior to the comparison of ratings for common rated products and vice versa.

Comparing Figure 7.28 and Figure 7.28, high visibility commonly requires the comparison of hypothesized preferences. The reason being is that the set of potential like-minded individuals is large. The comparison of ratings for common rated products is inappropriate due to the issue of partial preference representation.

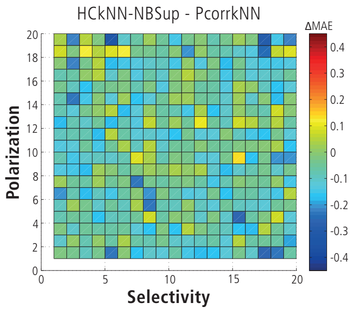
Likewise, low effort commonly requires the comparison of hypothesized preferences. The reason being is that machine learning efficiently generalizes the individuals preferences from few ratings. Thus, the comparison of hypothesized preferences mitigates the issues of similarity significance, partial preference representation and similarity assessability.



(a) Dataset with 0% sparsity.

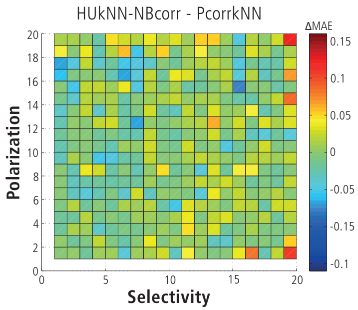


(b) Dataset with 40% sparsity.

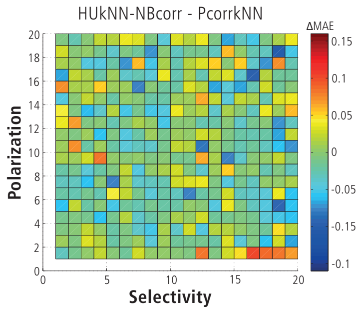


(c) Dataset with 80% sparsity.

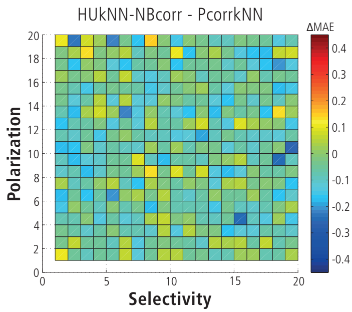
Figure 7.26: Comparison of recommendation performance and cold-start behavior of *HCkNN-NBSup* and PcorrkNN regarding individuals' selectivity and products' polarization.



(a) Dataset with 0% sparsity.



(b) Dataset with 40% sparsity.



(c) Dataset with 80% sparsity.

Figure 7.27: Comparison of recommendation performance and cold-start behavior of *HUkNN-NBcorr* and PcorrkNN regarding individuals' selectivity and products' polarization.

				Minor Cold-Start							
				Individual				Product			
				Effort		Selectivity		Visibility		Popularity	
				low	high	low	high	low	high	low	high
Minor Cold-Start	Individual	Effort	low	+					+		+
			high		-			-		-	
		Selectivity	low						+		
			high						+		
	Product	Visibility	low		-			-			
			high	+		+	+		+		
		Popularity	low		-						
			high	+							

(a) Grounded theory of hypothesis-based collaborative filtering for the minor cold-start problem.

Figure 7.28: Grounded theory of hypothesis-based collaborative filtering (HCF).

				Major Cold-Start							
				Individual				Product			
				Effort		Selectivity		Visibility		Popularity	
				low	high	low	high	low	high	low	high
Major Cold-Start	Individual	Effort	low	+				-	+		
			high								
		Selectivity	low						+		
			high								
	Product	Visibility	low	-				+			
			high	+		+			+		
		Popularity	low							+	
			high								

(b) Grounded theory of hypothesis-based collaborative filtering for the major cold-start problem.

Figure 7.28: Grounded theory of hypothesis-based collaborative filtering (HCF).

7.4 Theory Validation

We verified the grounded theory by conducting an empirical evaluation of its recommendation performance in terms of MAE and comparing it to HCF. We performed k-fold cross-validation using the same experimental settings as previously described in Section 6.1.

Instead of repeating the same evaluation as we described in Chapter 6, we selected either the predicted rating of the hypothesis-based collaborative filtering methods or the predicted rating of the baseline collaborative filtering method PcorrkNN. For this purpose, we considered exclusively the criteria from the grounded theory, as it is specified in Figure 7.28 in Section 7.3, to decide on a case-by-case basis from which method to take the predicted rating.

7.4.1 Experimental Setting

The grounded theory provides qualitative selection criteria. For this reason, we divided the 20 quantiles of individuals and products with respect to individual properties respectively product properties to three groups of equal size. We define the first six 20-quantiles (i.e., quantiles 1-6) as individuals respectively products having properties of low degree. Further, we define the last six 20-quantiles (i.e., quantiles 15-20) as individuals respectively products having properties with high degree. Finally, we define the remaining quantiles as individuals and products having properties with ordinary degree.

We applied the theory for minor cold-start situations, which we described in Figure 7.28 in the previous section, for datasets with sparsity 0% to 70%. From this, we derived the following selection criteria:

$$\widehat{r}_{ig} = \begin{cases} \widehat{r}_{ig} \text{ of PcorrkNN} & \text{i.effort} = \text{ high} \\ & \vee \text{ g.visibility} = \text{ low} \\ & \vee \text{ i.effort} = \text{ high} \wedge \text{ g.visibility} = \text{ low} \\ & \vee \text{ i.effort} = \text{ high} \wedge \text{ g.popularity} = \text{ low} \\ \widehat{r}_{ig} \text{ of HCkNN} \dots & \text{, otherwise} \end{cases} \tag{7.1}$$

We used the theory for major cold-start situations, which we described in Figure 7.28 in the previous section, for datasets with sparsity 70% to 90%. From this, we derived the following selection criteria:

$$\widehat{r}_{ig} = \begin{cases} \widehat{r}_{ig} \text{ of PcorrkNN} & \text{i.effort} = \text{ low} \wedge \text{ g.visibility} = \text{ low} \\ \widehat{r}_{ig} \text{ of HCkNN} \dots & \text{, otherwise} \end{cases} \tag{7.2}$$

7.4.2 Results and Discussion

We evaluated the rating prediction accuracy of the methods in terms of MAE. The MAE results of the evaluated methods are presented in Table 7.3. Figure 7.29 provides a visual representation of the MAE values for the better comprehension of the results. For readability reason, Figure 7.29 provides a selection of the evaluated methods that are presented in Table 7.3.

As shown in Table 7.3 as well as in Figure 7.29, all methods taking the grounded theory into account perform better with respect to the datasets with minor cold-start problem (i.e., 0% to 30% sparsity) relative to their equivalents not taking the grounded theory into account. In the case of major cold-start situation, the grounded theory does provide marginal improvement.

In the case of minor cold-start situation, the significantly best performing method is *HCkNN-NBSup*^{GT}, which combines *HCkNN-NBSup* and Pcor-

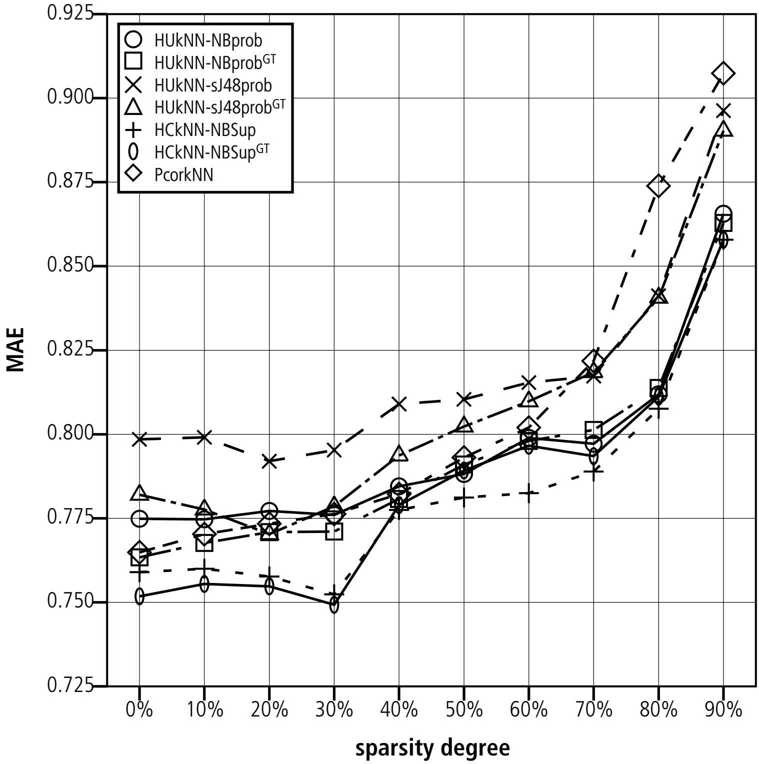


Figure 7.29: Behavior of recommendation performance in terms of MAE with increasing degree of sparsity from 0% (original data set) to 90%

Method	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**
HUkNN-J48corr	0.803	0.805	0.808	0.807	0.816	0.824	0.8300	0.839	0.859	0.900
HUkNN-J48corr^{GT}	0.785	0.781	0.785	0.784	0.795	0.804	0.811	0.842	0.8600	0.897
HUkNN-NBcorr	0.766	0.767	0.770	0.768	0.775	0.784	0.792	0.800	0.820	0.884
HUkNN-NBcorr^{GT}	0.756	0.760	0.765	0.764	0.773	0.785	0.794	0.803	0.819	0.876
HUkNN-SVMcorr	0.769	0.768	0.774	0.776	0.780	0.791	0.794	0.805	0.836	0.873
HUkNN-SVMcorr^{GT}	0.755	0.758	0.765	0.768	0.774	0.786	0.793	0.807	0.833	0.868
HUkNN-J48prob	0.789	0.788	0.790	0.790	0.796	0.800	0.802	0.807	0.829	0.876
HUkNN-J48prob^{GT}	0.777	0.773	0.778	0.778	0.786	0.797	0.804	0.809	0.828	0.871
HUkNN-NBprob	0.775	0.775	0.777	0.776	0.785	0.788	0.799	0.797	0.812	0.866
HUkNN-NBprob^{GT}	0.763	0.768	0.771	0.771	0.781	0.791	0.798	0.801	0.814	0.863
HUkNN-sJ48prob	0.799	0.799	0.792	0.795	0.809	0.810	0.815	0.817	0.841	0.896
HUkNN-sJ48prob^{GT}	0.782	0.778	0.770	0.779	0.794	0.802	0.810	0.819	0.841	0.890
HCkNN-J48NoG	0.786	0.788	0.791	0.797	0.794	0.801	0.803	0.820	0.845	0.922
HCkNN-J48NoG^{GT}	0.772	0.769	0.770	0.771	0.786	0.797	0.804	0.821	0.843	0.918
HCkNN-J48Sup	0.786	0.788	0.791	0.796	0.794	0.801	0.804	0.819	0.844	0.923
HCkNN-J48Sup^{GT}	0.772	0.769	0.770	0.771	0.786	0.797	0.804	0.820	0.843	0.918
HCkNN-NBSup	0.759	0.760	0.758	0.752	**0.778**	**0.781**	**0.783**	**0.789**	**0.808**	**0.858**
HCkNN-NBSup^{GT}	**0.752**	**0.756**	**0.755**	**0.749**	0.779	0.789	0.797	0.794	0.811	**0.858**
HCkNN-J48SSup	0.776	0.777	0.776	0.774	0.788	0.794	0.796	0.803	0.821	0.885
HCkNN-J48SSup^{GT}	0.766	0.765	0.766	0.762	0.785	0.795	0.803	0.807	0.823	0.881
HCkNN-NBSSup	0.825	0.823	0.815	0.806	0.841	0.841	0.840	0.845	0.852	0.885
HCkNN-NBSSup^{GT}	0.784	0.787	0.784	0.778	0.808	0.816	0.822	0.849	0.856	0.885
PcorrkNN	0.765	0.770	0.773	0.776	0.782	0.793	0.802	0.822	0.874	0.907

Table 7.3: Comparison of MAE between common hypothesis-based methods and hypothesis-based methods which take the grounded theory into account.

rkNN based on the criteria of the grounded theory. In conclusion, taking the grounded theory into account provides a general improvement and therefore verifying the grounded theory.

Nevertheless, the recommendation performance in between drops substantially. However, the recommendation performance can be improved by adjusting the selection criteria from Eq. (7.1) or define a different classification of the quantiles.

7.5 Acceptance of Hypotheses

We used grounded theory methodology and developed a theory to explain the cold-start behavior of HCF. We defined six concepts to explain the theory. In the following, we discuss the acceptance of Hypothesis H4 (rating predicate hypothesis).

Rating predicate hypothesis (H4). We defined six concepts based on the ratings which are provided by individuals and received by products. We have shown that this six concepts strongly correlate to the performance difference of collaborative filtering methods, which retrieve like-minded individuals to provide recommendations. More precisely, we compared the performance difference of methods based on retrieving like-minded individuals based on the similarity of their hypothesized preferences and methods based on the similarity of ratings for common rated products.

We can explain the cold-start behavior of hypothesis-based collaborative filtering based on our theory. This theory is appropriate to forecast if the recommendations should be based on the similarity of hypothesized preferences or on the similarity of ratings for common rated products.

Hence, we accept H4.

V
Closing

8

Limitations

HYPOTHESIS-BASED collaborative filtering retrieves like-minded individuals based on the similarity of individuals' hypothesized preferences. This method relies on the accuracy of hypothesized preferences, which causes technical limitations. Besides that, retrieving like-minded individuals based on the similarity of both hypothesized preferences has further limitations.

In the following, we discuss the conceptual limitations in Section 8.1 and the technical limitations in Section 8.2

8.1 Conceptual Limitations

As shown in Chapter 6, the accuracy of individuals' hypothesized preferences depends on the number of ratings they provide. Generally, the more ratings individuals provide, the more accurate are their respective hypothesized preferences. In contrast, the less ratings individuals provide, the less accurate are their respective hypothesized preferences.

As shown in Chapter 7, the comparison of inaccurate hypothesized preferences provides inaccurate similarities, thus individuals getting mistaken as being like-minded as well as like-minded individuals getting overlooked.

However, the comparison of inaccurate hypothesized preferences is superior to the comparison of ratings for common rated products regarding the selection of like-minded individuals from a large set of individuals, especially in major cold-start situations.

The accuracy of hypothesized preferences is limited by the hypothesis space, which is human designer's choice. For that reason, machine learning algorithms are not able to find more accurate hypothesized preferences for individuals which provide many ratings already. Therefore, the similarity accuracy of hypothesized preferences is limited. As shown in Chapter 7, retrieving like-minded individuals which provide many ratings based on the similarity of their respective hypothesized preferences is inferior to the comparison of the ratings for common rated products.

In the case of products which have been rated by few individuals, the limited accuracy of hypothesized preference similarity results in inferior recommendation performance relative to the accuracy of the rating similarity for common rated products.

8.2 Technical Limitations

The *hypothesized utility-based preference similarity* (HU preference similarity) of two individuals' hypothesized preferences is computed based on a set of products, which we discussed in Section 5.2.1. We suggest to compute the hypothesized preference similarity on the basis of the unified product set of both individuals to mitigate the issues we discussed in Section 1.1. However, the unified product set of two individuals may be biased due to two reasons. Firstly, both individuals, typically, provide different amount of ratings, thus shifting the comparison of hypothesized preferences towards the products which the individual with the most ratings has rated. To overcome this bias, we suggest to take the amount of ratings of individuals into account and weight the similarity of hypothesized utilities accordingly.

Secondly, hypothesized preferences predict the same utility for sufficient similar products and hence, the similarity of hypothesized utilities remains the same. For this reason, sufficient similar products bias the comparison of hypothesized utilities and shifts the comparison towards these products. To overcome this bias, we suggest two approaches. The first approach is to take the redundant information of sufficient similar products into account by means of weighting the similarity of hypothesized utilities accordingly. The second approach is to take the information gain of the comparison of products' utilities into account. For instance, products and product properties which provide high utility to most individuals provide low information gain, thus biasing the comparison of hypothesized preferences.

The *hypothesis composition-based preference similarity* (HC preference similarity) of two individuals' hypothesized preferences is computed based on the sets of *hypothesized partial preferences* (HPPs) which are extracted from the corresponding machine learning models. Different machine learning algorithms represent differently individuals' hypothesized preferences, thus resulting in different representation of HPPs. For this reason, HC preference similarity is limited to the comparison of two different machine learning models which use the same representation.

Furthermore, the frequency with which partial preferences are addressed by products is not considered by the HC preference similarity method. Hence, partial preferences which are rarely addressed highly influence the comparison of two individuals' hypothesized preferences relative to their frequency.

To compute the HC preference similarity of two individuals' hypothesized preferences, each HPP of one individual is compared to each HPP of the other individual. Hence, the computational effort of HC preference similarity depends on the one hand on the set sizes of both corresponding HPPs and the complexity of HPPs, which primarily depends on the size of the hypothesis space.

SemTree uses a domain ontology to generalize more efficiently from observations by considering the semantic relationship among concepts. In the context of product properties, no semantic relationships exists in many cases among product properties, however.

The foundation of both presented algorithmic frameworks are the individuals' hypothesized preferences. The accuracy of the comparison of two individuals' hypothesized preferences relies on the accuracy of both hypothesized preferences. Therefore, the less accurate both hypothesized preferences are the less accurate is the comparison of both hypothesized preferences.

The accuracy of hypothesized preferences depends on the hypothesis space, which limits the accuracy of individuals' hypothesized preferences, and the number of ratings which are provided by the corresponding individual. A trade-off exists between a large hypothesis space and individuals' hypothesized preferences being partial functions and, thus, not able to hypothesize the utility of every product. This trade-off needs to be taken into account comparing hypothesized preferences, especially in the cold-start situation.

This limitation could be addressed by incorporating domain ontologies which define semantical relationships among product properties, thus providing additional information to the used machine learning algorithm. SemTree (see Chapter 4) is an example of using domain ontologies to generalize more efficiently from individuals' ratings to the individuals' preferences. An additional source of ontological knowledge about items is the Linked Data Cloud, which could be exploited analogously or even opportunistically to describe specific products in more detail or gather missing information.

The recommendation coverage of *hypothesis-based collaborative filtering* (HCF) is limited to products which have been rated at least once. However, the comparison of two individuals' hypothesized preferences allows for the comparison of two individuals which do not necessarily share ratings for the same products. In contrast, the comparison of two individuals based on the ratings for common rated products individually limits the recommendation coverage even more.

9

Conclusions

RETRIEVING like-minded individuals by comparing their ratings for common rated products is oftentimes difficult as the rated products are not necessarily a representative sample of individuals' preferences. There are four reasons. Firstly, the set of common rated products is too sparse to draw a significant conclusion about the preference similarity of both individuals.

Secondly, ratings for common rated products correspond to the intersection of two individuals' rated products and thus may represent only partially both individuals' preferences. Consequently, overall preference similarity is, in fact, deduced from partial preference similarity.

Thirdly, the preference similarity between two individuals is not assessable in the case when both individuals do not share ratings for the same products. Consequently, like-minded individuals are missed due to lack of ratings.

Lastly, retailers collect only a fraction of individuals' ratings on their store, because individuals purchase products from different stores. Hence, individuals' ratings are distributed across multiple retailers, which limits the set of common rated products per retailer.

Counteracting these problems, we presented in this dissertation *hypothesis-*

based collaborative filtering (HCF). The idea is to hypothesize individuals' preferences and compare their respective hypothesized preferences instead of ratings for common rated products.

To this goal, we provide the theoretical foundation to model individuals' preferences by means of hypothesized preferences. Furthermore, we introduce the notion of partial preferences to allow for fine-grained comparison of individuals' preferences. Finally, we provide two techniques to extract *hypothesized partial preferences* (HPPs) from machine learning models as well as the preference ontology YOULIKE to specify these preferences. Additionally, we provide the machine learning algorithm SEMTREE, which uses a domain ontology to boost the efficiency of learning from observations and which provides more accurate hypothesized preferences for individuals with few ratings.

We provide the theoretical foundation of *hypothesized preference similarity*, *hypothesized partial preference similarity* (HPP similarity) and *hypothesized semi-partial preference similarity* (HSPP similarity). Based on this, we provide two different algorithmic frameworks to compare hypothesized preferences. Firstly, we provide an algorithmic framework to compare the similarity of hypothesized preferences based on the predicted utilities some products provide to individuals. We present two methods to compare individuals' preferences. Furthermore, we provide a method to compute the HSPP similarity of individuals.

Secondly, we provide an algorithmic framework to compare the similarity of hypothesized preferences based on the comparison of the composition of hypothesized preferences. To this goal, we propose amongst other things a semantic extension of the Jaccard similarity coefficient to compute the similarity of HPPs by means of the semantic relations among sets of product features.

We empirically evaluated our proposed algorithmic frameworks on the real world MovieLens 100k dataset and compared it to state-of-the-art collaborative filtering methods. Furthermore, we evaluated the cold-start behavior by reducing gradually the number of ratings of individuals in the dataset. Our empirical study provides significant evidence for the robustness of HCF against data sparsity and the superiority to state-of-the-art collaborative filtering methods, especially regarding major cold-start situations.

We used grounded theory methodology to scrutinize the data of the empirical study to understand the phenomenon of HCF outperforming other collaborative filtering methods. We show that especially in the case of individuals providing few ratings the comparison of hypothesized preference to retrieve like-minded individuals is superior to the comparison of ratings for common rated products. Furthermore, we show that the comparison of hypothesized preferences is more effective in retrieving like-minded individuals from a large sets of individuals

Finally, we provide the RECOMIZER framework, which, amongst other things, provides the Java implementations of our proposed algorithmic framework and exploits the computational power of multi-core processor systems. In addition, we provide the movie recommender system OMORE, which provides individuals recommendations across retailers.

In conclusion, our findings help improving automated recommendations both on the traditional, human and the Semantic Web. As such, our approach provides an important building block for building next-generation recommender systems to further increase consumers welfare and thus ultimately increasing economic and social welfare.

9.1 Acceptance of Hypotheses

For the fulfillment of our research goals (see Section 1.3.2), we formulated seven hypotheses (see Section 1.3.1), which we have verified in this dissertation. We briefly discuss wether we accept or reject these hypotheses:

Information gain of domain ontology hypothesis (H1)

accepted.

We demonstrated and discussed by means of an example in Section 4.2.2 how a domain ontology can help in the context of machine learning to improve the efficiency of generalizing from few observations. We showed that, for instance, a taxonomic relationships among feature concepts, which describe products, provide a better interpretation of individuals' ratings, thus generalizing more efficiently from individuals' ratings to the individuals' preferences.

Since domain ontologies can provide additional information which makes machine learning more efficient, we conclude that, besides of observations, machine learning gains information from domain ontologies.

Therefore, we accept Hypothesis H1.

Ontology-based preference representation hypothesis (H2):

accepted.

Based on the theoretical foundation (see Section 3.1) and the notion of partial preferences (see Section 3.1.1), we demonstrated how to extract HPPs from two different representations of hypotheses, namely decision tree models (see Section 3.2.1) and Naïve Bayes models (see Section 3.2.2). Based on the preference ontology YOULIKE, we demonstrated how to specify hypothesized preferences (see Section 3.3).

Since we can specify hypothesized preferences with the preference ontology YOULIKE, we conclude that it is feasible to specify hypothesized preferences with ontologies in general, the preference ontology YOULIKE in particular.

Therefore, we accept Hypothesis H2.

Hypothesized utility-based preference similarity hypothesis (H3.1):

accepted.

With reference to the empirical study regarding rating prediction accuracy (see Section 6.4.1), retrieving like-minded individuals based on the HU preference similarity framework outperforms baseline methods, especially when the sparsity degree gets higher. Furthermore, this method provides similar performance as retrieving like-minded individuals based on ratings for common rated products.

Regarding the relevance filtering quality, the methods based on HU preference similarity framework outperforms the similar baseline method PcorrkNN.

Since the methods based on our HU preference similarity framework outperforms baseline methods, especially in terms of rating prediction accuracy, we conclude that it is appropriate to retrieve like-minded individuals. In fact, it may even better retrieve like-minded individuals.

Therefore, we accept Hypothesis H3.1.

Hypothesis composition-based preference similarity hypothesis (H3.2):

accepted.

With regards to rating prediction accuracy, retrieving like-minded individuals based on the HC preference similarity framework outperforms baseline methods, particularly when the sparsity degree gets higher. Furthermore, this method provides similar or better performance than retrieving like-minded individuals based on ratings for common rated products. Especially, *HCkNN-NBSup* outperforms all other methods in almost any case.

Regarding to the relevance filtering quality, the methods based on HC preference similarity framework outperforms baseline methods in almost any case, *HCkNN-NBSup* in particular.

Since our proposed HC preference similarity framework outperforms other baseline methods, PcorrkNN in particular, we conclude that it is appropriate to retrieve like-minded individuals. In fact, it retrieves better like-minded individuals, *HCkNN-NBSup* in particular.

Therefore, we accept Hypothesis H3.2.

Hypothesized partial preference similarity hypothesis H3.3:

not accepted.

Regarding to rating prediction accuracy, considering HPPs or HSPPs to retrieve like-minded individuals does in some cases outperform baseline collaborative filtering methods, particularly in major cold-start situations.

Regarding to the relevance filtering quality, the method *HUkNN-sJ48prob*, which considers HPP similarity, did not outperform baseline collaborative filtering methods in general. For that reason, we conclude that *HUkNN-sJ48prob* does not retrieve accurately enough like-minded individuals.

Considering rating prediction accuracy and relevance filtering quality, we conclude that considering HPP similarity or HSPP similarity is not accurate enough to retrieve like-minded individuals. Nevertheless, we think that this method may be appropriate when considering in addition overall similarity.

Hence, we do not accept Hypothesis H3.3.

Cold-start mitigation hypothesis H3.4:

accepted.

With regards to rating prediction accuracy, retrieving like-minded individuals by comparing the similarity of hypothesized preferences significantly and substantially outperformed baseline methods in cold-start situations, *HCkNN-NBSup* in particular.

Regarding to the relevance filtering quality, our methods generally outperformed PcorrkNN in major cold-start situations. However, the baseline collaborative filtering methods SVD and WoC performed significantly best in cold-start situations. Nonetheless, both baseline methods do not retrieve like-minded individuals to filter products. Thus, we argue that HCF perform significantly best in cold-start situations in terms of retrieving like-minded individuals.

We conclude that HCF mitigates the cold-start problem since the retrieval of like-minded individuals based on the comparison of the similarity of hypothesized preferences outperforms the relevant baseline methods.

Therefore, we accept Hypothesis H3.4.

Rating predicate hypothesis H4:

accepted.

We defined six concepts based on the ratings which are provided by individuals and received by products. We have shown that this six concepts strongly correlated to the performance difference of collaborative filtering methods which retrieve like-minded individuals to provide recommendations. More precisely, we compared the performance difference of collaborative filtering methods which retrieve like-minded individuals based on the similarity of their hypothesized preferences and based on the similarity of ratings for common rated products, respectively.

We can explain the cold-start behavior of HCF based on our theory. This theory is appropriate to forecast if the recommendations should be based on the similarity of hypothesized preferences or on the similarity of ratings for common rated products.

Therefore, we accept Hypothesis H4.

9.2 Achievements of Research Goals and Thesis

In this section, we summarize the justification of the fulfillment of our research goals and the verification of the thesis.

The presented machine learning algorithm SEMTREE increases the efficiency of hypothesized individuals' preferences using a domain ontology. The success of this research goal depends on the acceptance of Hypothesis H1. Since we accepted Hypothesis H1, we have fulfilled Research Goal G1.

The presented preference ontology YOULIKE is appropriate to describe individuals' hypothesized preferences. The success of this research goal depends on the acceptance of Hypothesis H2, which we accepted. Therefore, we have fulfilled Research Goal G2.

Both presented algorithm framework provide the foundation of several HCF methods (e.g., HypokNNNBCompSup). These HCF methods (e.g., *HCkNN-NBSup*) are superior to the compared collaborative filtering methods, particularly in the case of major cold-start situations. The success of this research goal depends on either acceptance of either Hypotheses H3.1, H3.2, or H3.3. Since we have accepted Hypothesis H3.1 and Hypothesis H3.2, we have fulfilled Research Goal G3.

We conducted an empirical study and provide significant evidence that HCF based on both algorithmic frameworks are superior to the compared collaborative filtering methods. The success of this research goal depends on the acceptance of either Hypotheses H3.1, H3.2 and H3.3, and the acceptance of the Hypothesis H3.4. Since we accepted Hypotheses H3.1, H3.2 and H3.4, we have fulfilled Research Goal G4.

We developed a theory which explains the phenomenon of HCF outperforming other collaborative filtering methods. The theory can be used to forecast which method provides more accurate recommendations depending on the context. The success of this research goal depends on the acceptance of the Hypothesis H4. Since we accepted Hypothesis H4, we have fulfilled Research Goal G5.

We fulfilled all research goals. Therefore, we regard the thesis of this dissertation as verified.

9.3 Opportunities for Future Research

In this dissertation, we provide two different algorithmic frameworks to compute the similarity of individuals' hypothesized preferences and provide evidence of their effectiveness in the context of recommender systems, collaborative filtering in particular. We used the research methodology of grounded theory to analyze and explain the cold-start behavior of HCF.

In Section 3.1.1, we introduced the notion of HPPs, which constitutes individuals' hypothesized preferences. The comparison of HPPs plays a fundamental role in HC preference similarity. However, we have not investigated the frequency distribution of HPPs over all hypothesized preferences and their interrelationship or co-existence. We think that conducting further research in this area may lead to better understanding of the diversity of individuals' preferences to ultimately improve recommender systems. More specifically, we assume that that some HPPs are more common than others. In addition, we assume that some HPPs may be frequently respectively rarely part of the same hypothesized preferences, thus correlating. Hence, the correlation of HPPs can be used to extend incomplete hypothesized preferences with likely HPPs, especially in major cold-start situations.

Additionally, the frequency distribution of HPPs and their interrelationship in terms of correlation may reduce the computational effort to retrieve like-minded individuals. We think that the comparison of hypothesized preferences can be reduced to the comparison of HPPs which provides the highest information gain.

Finally, we think that the investigation of the distribution and the interrelationship or co-existence of HPPs is a promising research direction which may provide further insights into the principle of collaborative filtering and moreover insights into the multifaceted nature of individuals' preferences.

We introduced the notion of HPP similarity and HSPP similarity in Section 5.1.1 and Section 5.1.2 respectively. In Chapter 6, we provide evidence of the feasibility of retrieving like-minded individuals based on the HSPP similarity. Although this method is effective, it does not outperform other methods in terms of recommendation performance. However, we think that a combination of partial preference similarity and overall preference similarity may improve recommendation performance and provide plausible explanations for recommendations, thus representing a promising direction for further research.

In this dissertation, we provide two algorithmic frameworks to compare hypothesized preferences. Although this dissertation contributes primarily to the research field of recommender systems, both algorithmic framework may contribute to further research fields by enabling the comparison of prediction models. For example in [Zenger et al., 2011], we predict software defaults or rather bugs in computer programs to help software developers find and resolve them more quickly and with lower costs. To predict bugs in a software project, we use its bug prediction models and compare it to other bug prediction models of other software projects. As in collaborative filtering, we combine the corresponding most similar bug prediction models of other projects to predict bugs in the software project.

VI
Appendix

A

Tools

In the following, we present the tools that we developed for this dissertation.

A.1 RECOMIZER

We developed RECOMIZER to verify our thesis and to evaluate the recommendation performance of both presented algorithmic frameworks (i.e., HU and HC preference similarity) to retrieve like-minded individuals. RECOMIZER provides a Java implementation of several collaborative filtering methods, which are based on both algorithmic frameworks and which differ, amongst other things, regarding machine learning algorithms. Furthermore, it provides implementations of several recommender systems from the related work. We applied RECOMIZER to perform the empirical study presented in Chapter 6 and Chapter 7.

RECOMIZER supports multi-core processor systems. The computation of hypothesized preference similarity of individuals is distributed among available cores to reduce the computation time.

RECOMIZER reuses the WEKA library [Witten and Frank, 2005] to hypothesize individuals' preferences with machine learning. In RECOMIZER, instances of individuals are dynamically decorated with the functionality of hypothesized preferences. This enables the comparison of individuals' hypothesized preferences. To extract the hypothesized partial preferences, we developed a BNF and use *javacc* to parse the textual representation of machine learning models.

The domain model of RECOMIZER consists mainly of individuals, products and ratings of individuals for products. The Java Persistence API (JPA) is used by RECOMIZER to persist instances of the domain in a relational datastore.

The RECOMIZER framework is depicted as source code map in Figure A.1.

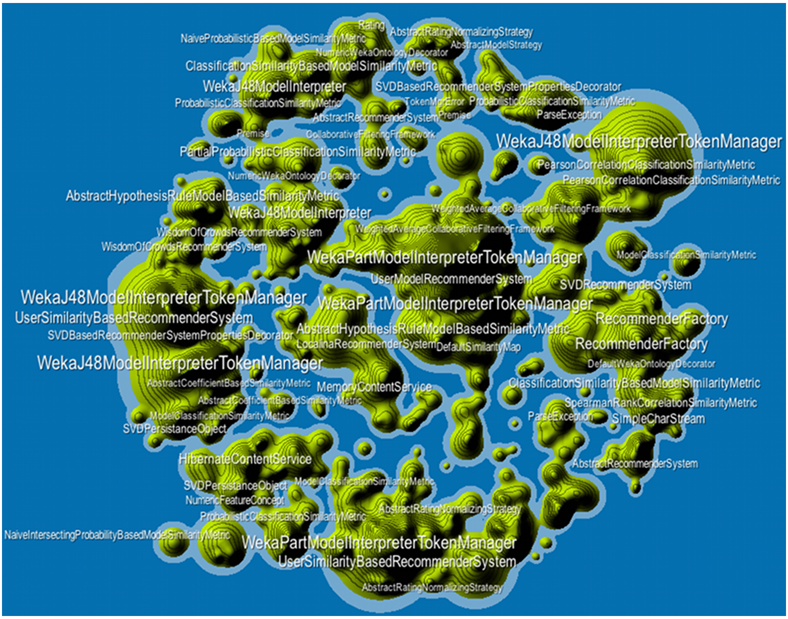


Figure A.1: Source code map of Recomizer-Core.

A.2 OMORE

Recommender systems face two main problems. First, ratings which individuals expressed in one online store or portal are stored on server side and cannot be reused on other Web applications. Referring to the issue of preferences incompleteness, recommender systems have only partial knowledge about individuals' preferences. Furthermore, individuals have to rate the same products over and over again on each Web site they want to take advantage from recommendations.

Secondly, individuals have to trust the owner of the Web application, that his data is protected from unauthorized access and it is not misused and shared with third parties.

To overcome the issue of preferences incompleteness, we developed OMORE. OMORE is a personal movie recommender, which runs locally as a Web browser extension, specifically a Firefox[1] add-on, and lets individuals rate movies when they browse supported Web pages.

The ratings are stored locally which tackles the privacy issue. The ratings are used to learn the individuals' preference model. In the configuration plane of the add-on, individuals can choose among different machine learning algorithms. The individuals's hypothesized preferences are then used to generated personal recommendations which are directly visible within the supported Web sites which individuals are currently browsing.

To enable the cross-referencing between these movie sites, we use our movie ontology MO (see Appendix B) and URIs to uniquely identify movies across different movie sites that is key to provide cross-page recommendations.

[1] `http://www.mozilla.org/firefox/`

A.2.1 Architecture

The architecture of the OMORE is presented in Figure A.2. The central part of OMORE is the machine-learning component. It consists of the Learner, which is responsible for learning the individual model and the Classifier for generating recommendations based on the individual model [Basu et al., 1998]. An individual can choose between several machine learning algorithms from the WEKA library [Witten and Frank, 2005].

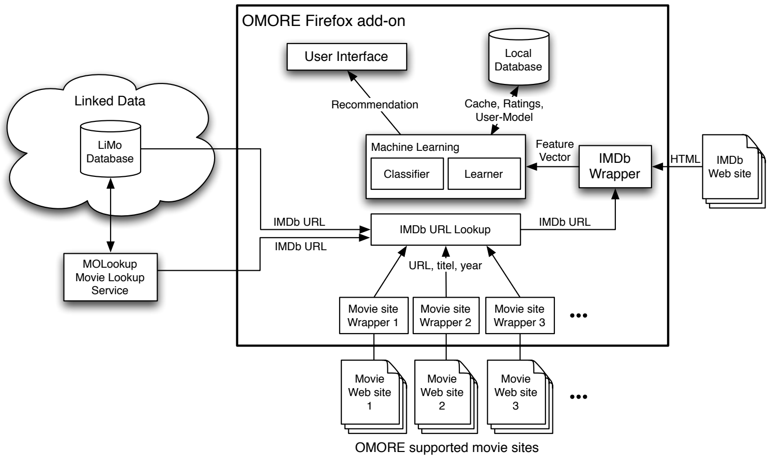


Figure A.2: Overview of the OMORE Architecture.

When individuals browse a Web page, the URL is checked by the OMORE Firefox add-on if the current URL is a movie page that is supported by OMORE. This is done by analyzing the URL of the Web page. The URL also indicates which Movie Site Wrapper should be used. The wrapper extracts the URL, the title, and release year of the movie. This information is then used to retrieve the IMDb URL from the LiMo database or the MOLookup service (see section above). The URL is then handed over to the IMDb Wrapper to build the feature vector.

When individuals rate a movie, the feature vector and the rating are stored in the local SQLite database that comes with Firefox. When individuals have rated enough movies (threshold can be set in the configuration), the learner retrieves the feature vectors and the individuals' ratings from the Local Database and starts the learning of their preference models.

Generally, the learning of a model is computationally expensive in contrast to the use of the model. Therefore, we store the learned individual model locally and reuse it for the generation of new recommendations. This way, OMORE scales over the movies rated by individuals. The preference models are recomputed in regular intervals to improve the adaptation to the individuals' preferences. The feature vector is also used by the Classifier to generate the recommendation which is displayed to the individual.

The wrapper is not only responsible for extracting information from the HTML Web page, but also for displaying the OMORE individual interface. The individual interface is integrated in the movie page by manipulating the DOM tree of the page. It displays the rating and gives the possibility to rate the movie. A screenshot is depicted in Figure A.3. OMORE can be easily extended to support additional movie pages by simply adding a new Movie Site Wrapper.

A.3 MOLookup

Establishing the links among movies from different Web pages is challenging due to the heterogeneous presentation of movies and the fact that Web pages may misspell, transform or extend movie titles in various ways. Especially on online shops, we experienced that the movie titles are extended with information about many variants of special or collector's edition and the type of medium the movie is provided. Instead of trying to extract the original title from the unpurified title, we decided to apply fuzzy search over movie titles and release year to retrieve the linkage among identical

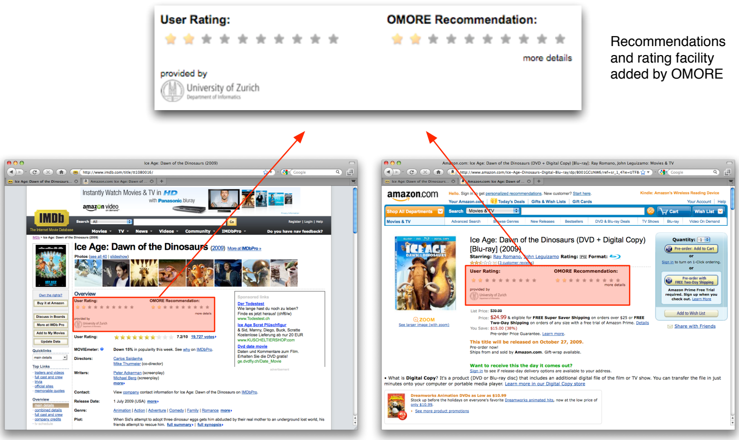


Figure A.3: Dynamically injected rating feature and recommendation of the OMORE element in two different Web sites.

movies from different Web pages.

For this purpose, MOLookup[2] uses the fuzzy search facility of Apache Lucene[3]. Lucene is a scalable and popular open-source search software that enables fast search within textual data. Lucine uses the Levensthein distance to compute the similarity between the given title and the titles stored in the LiMo database. Besides the JSON interface which is used by OMORE, MOLookup also provides a Web interface, which is depicted in Figure A.4

[2] http://seal.ifi.uzh.ch/molookup/
[3] http://lucene.apache.org

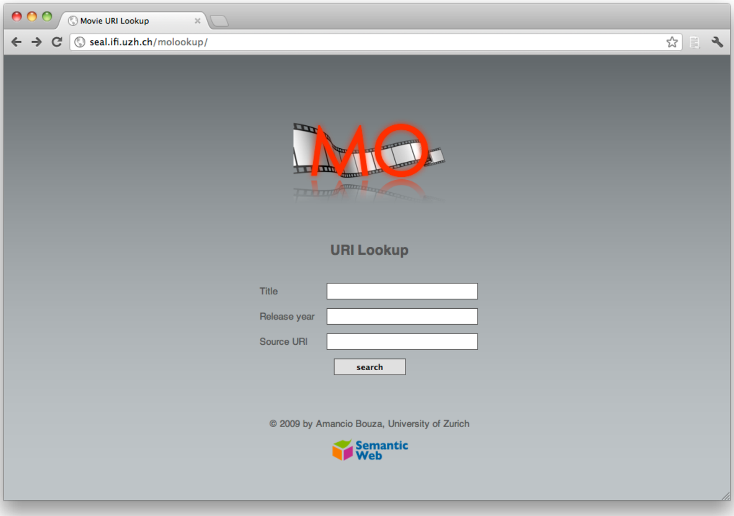


Figure A.4: Web interface of MOLookup.

A.4 LiMo Database

A cross-page movie recommender system also requires cross-references between the same movie across different Web sites. A first attempt exists in form of the Linked Movie Database[4] (LinkedMDB), but the available data is not as comprehensive as required for cross-page recommender systems such as OMORE. Thus, we came up with another approach to establish the link between identical movies across Web sites.

[4] `http://www.linkedmdb.org/`

A.4.1 Interlinking Movies across Web Pages

We use crowd-sourcing to establish the link between identical movies across Web sites and store these cross-references in our LiMo database[5]. Our LiMo database is based on D2R [Bizer and Cyganiak, 2006]. The LiMo database provides a unique URI for each movie, the URL to the IMDb movie page, the movie title, the release year, and the cross-references to all movie sites that are supported by OMORE.

OMORE uses the URL of the currently viewed Web page as key to look up the IMDb URL in the LiMo database. As discussed above, the IMDb URL is used to build the feature vector of a movie. When individuals browse a movie Web page where the cross-reference cannot be found in our LiMo database, OMORE uses our movie lookup service MOLookup[6] to retrieve the IMDb URL. OMORE uses the movie title, release year, and the URL of the Web page for the request to MOLookup. This information is extracted by OMORE from the currently viewed Web page. The title and the year are used for the lookup. The URL is used to store the new cross-reference in the LiMo database.

This way the LiMo database learns about the existence of a new movie page and stores the cross-reference. Individuals benefit from movie recommendations and we gain a comprehensive set of movie cross-references. This approach seems to be promising since it follows the Pareto improvement principle [Bernanke and Frank, 2007].

As an initial dataset for the LiMo database, we added all movies from the IMDb and the movie titles, release years, and IMDb URLs (about 1.4 million movies). In addition we have about 12 000 cross-references to Rotten Tomatoes and Amazon.com pages. The cross-references to the other supported movie pages (Netflix, Jinni, Blockbuster, LinkedMDB) will be established as discussed previously, when OMORE is used by the community.

[5] `http://seal.ifi.uzh.ch/limo`
[6] `http://seal.ifi.uzh.ch/molookup`

B

Movie Ontology MO

The movie ontology MO[1] aims to provide a controlled vocabulary to semantically describe movie related concepts such as movie, genre, director, actor and individuals such as "Ice Age", "Drama", "Steven Spielberg" or "Johnny Depp". We use the Web Ontology Language (OWL) to specify the MO ontology and to provide the needed ontology to describe movies applying Semantic Web technology. Hence, movies are semantically interpretable by individuals and software agents. We considered and integrated several related ontologies from the Linked Data Cloud and several other ontologies that are provided in the Linked Data cloud to highly couple the MO ontology with the Linked Data cloud to take advantage of synergy effects. The taxonomy of movie genres is depicted in Figure B.1.

The movie ontology MO was motivated by the fact that most movie ontologies specify superficially concepts, instances, and the semantic relations among these concepts. We think that a semantic movie ontology should provide on one hand concept hierarchies (e.g. for movie categorization and navigation support) and a sufficient set of individuals that can be used to describe movies. This enables user-friendly presentation of movie descriptions in the appropriate detail and the ability to describe movies taking advantage of the selection in controlled vocabulary of the MO ontology.

[1]We provide the movie ontology MO at `http://www.movieontology.org`.

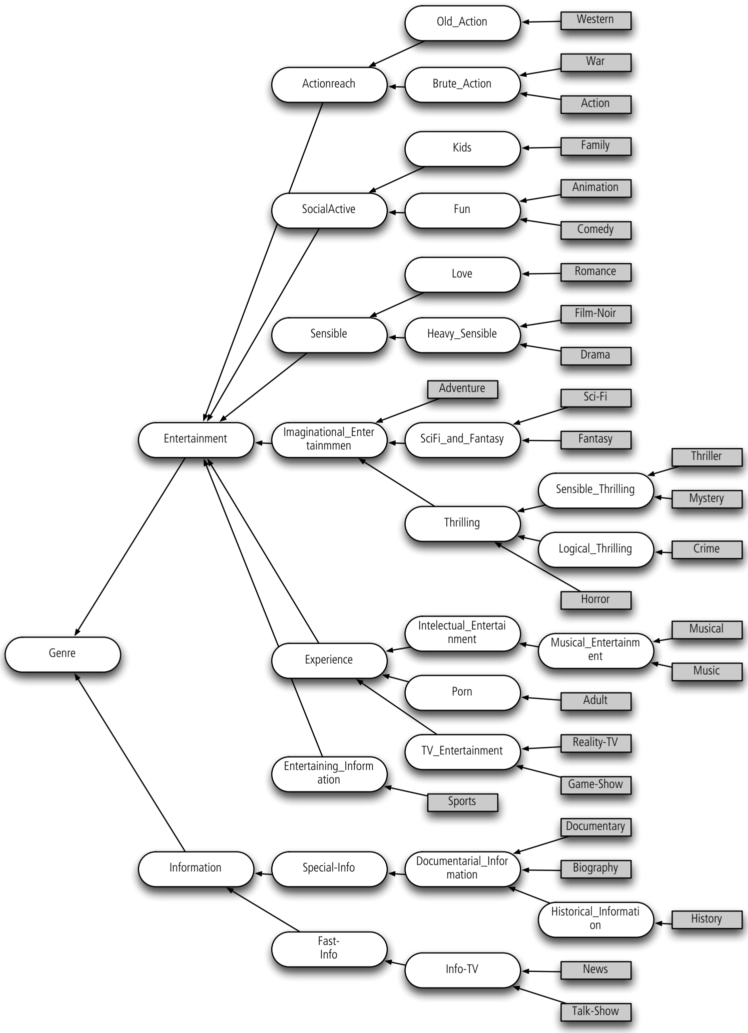


Figure B.1: Taxonomy of movie genres in the movie ontology MO.

C

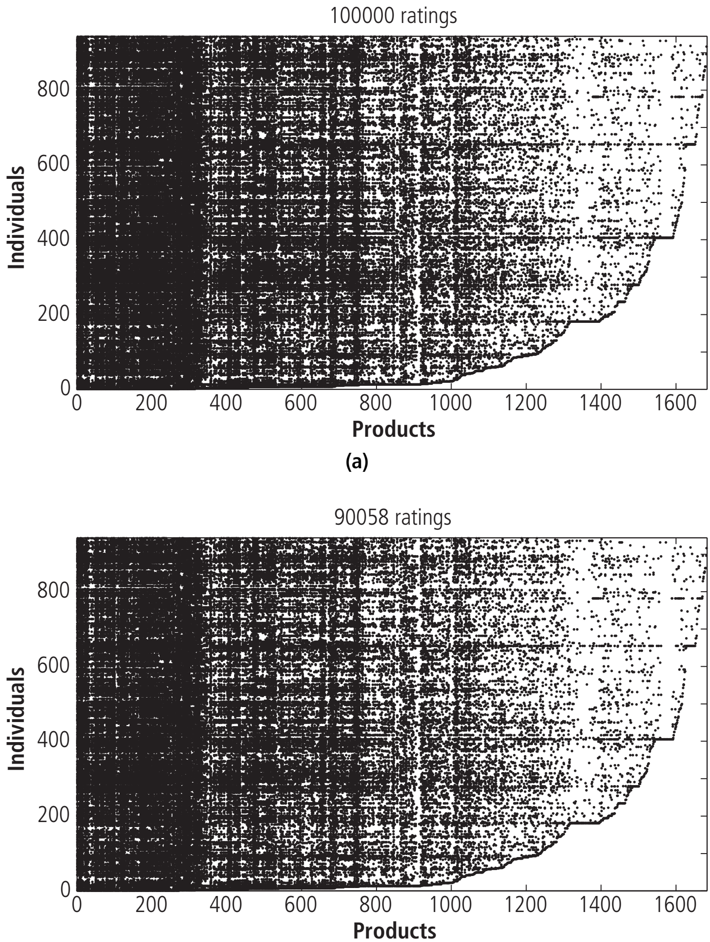
MovieLens Dataset

C.1 Genres of MovieLens

MovieLens Genre	MO Genre
unknown	—
Action	mo:Action
Adventure	mo:Adventure
Animation	mo:Animation
Children's	mo:Children
Comedy	mo:Comedy
Crime	mo:Crime
Documentary	mo:Documentary
Drama	mo:Drama
Fantasy	mo:Fantasy
Film-Noir	mo:Film-Noir
Horror	mo:Horror
Musical	mo:Musical
Mystery	mo:Mystery
Romance	mo:Romance
Sci-Fi	mo:Sci-Fi
Thriller	mo:Thriller
War	mo:War
Western	mo:Western

Table C.1: Genres from the MovieLens and their correspondence in the MO ontology.

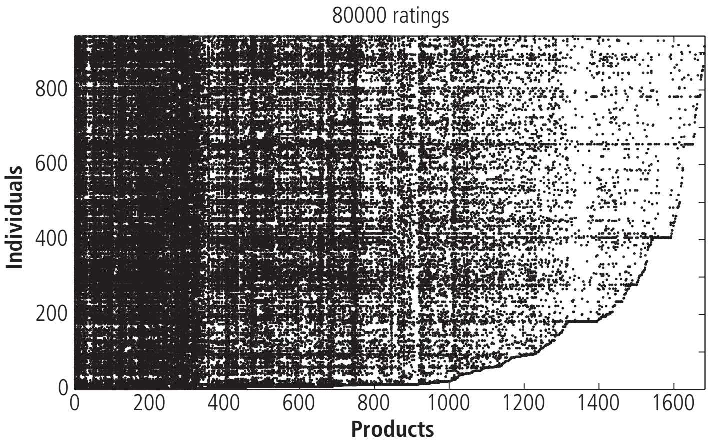
C.2 Sparse MovieLens Dataset



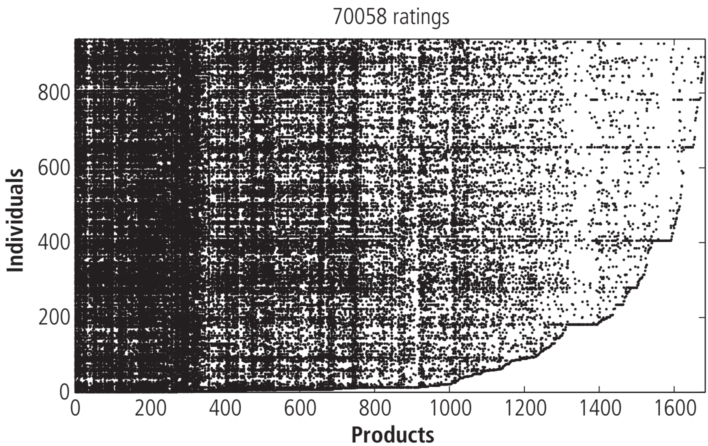
(a)

(b)

Figure C.1: Dataset collection with gradually increased degree of sparsity.

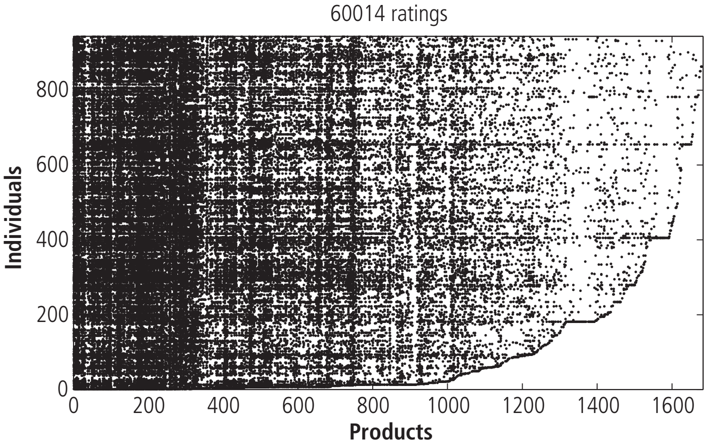


(c)

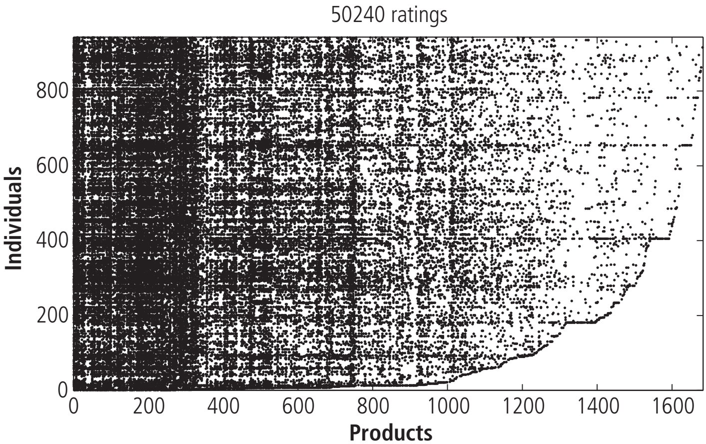


(d)

Figure C.1: Dataset collection with gradually increased degree of sparsity.

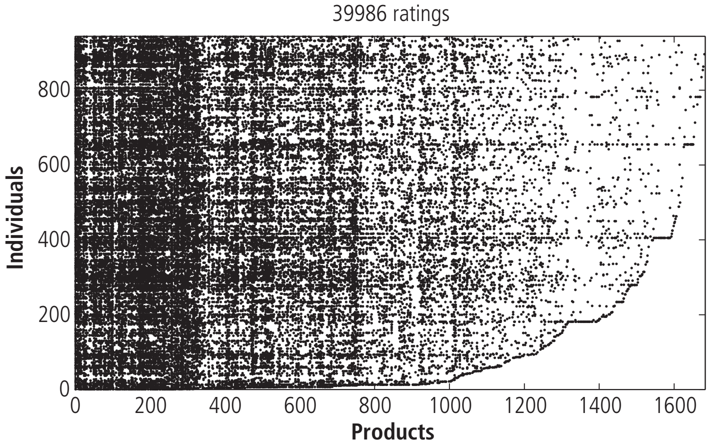


(e)

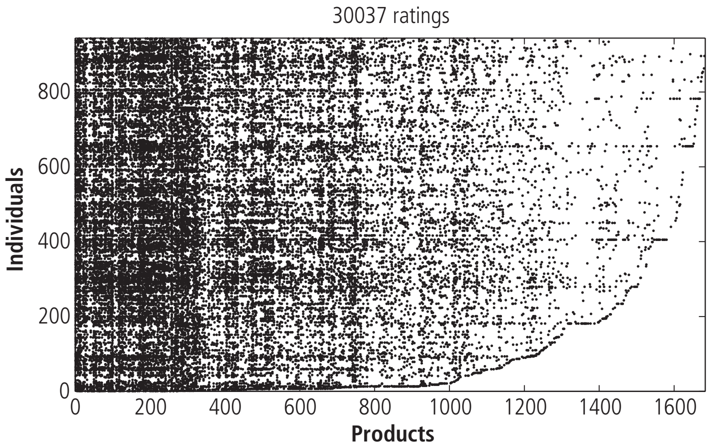


(f)

Figure C.1: Dataset collection with gradually increased degree of sparsity.

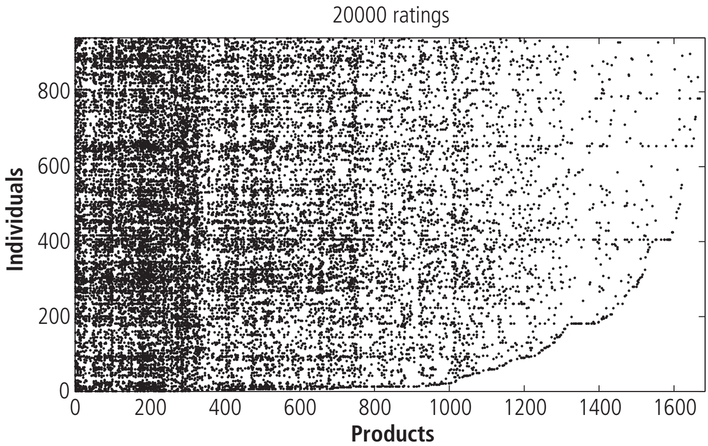


(g)

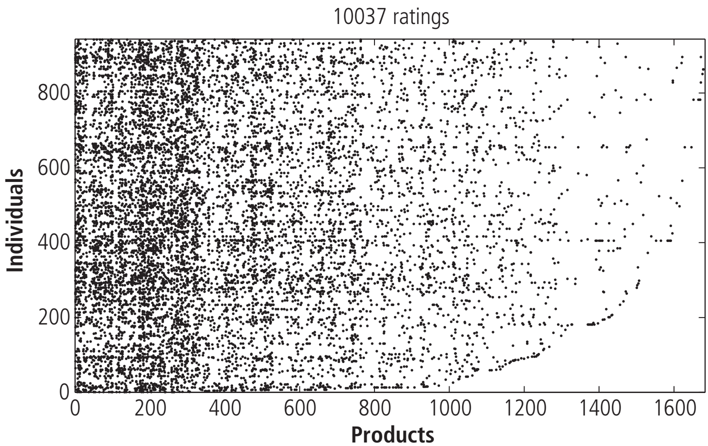


(h)

Figure C.1: Dataset collection with gradually increased degree of sparsity.



(i)

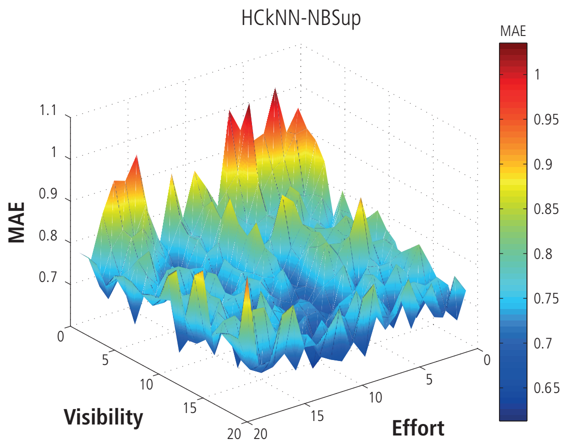


(j)

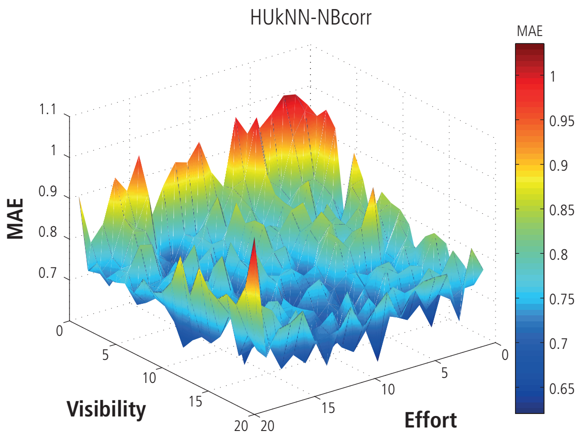
Figure C.1: Dataset collection with gradually increased degree of sparsity.

D

Distribution of Recommendation Performance

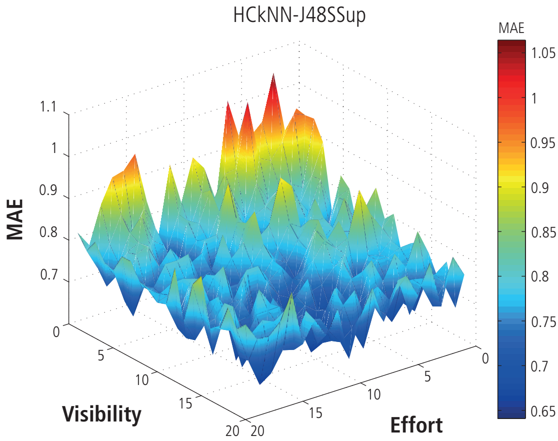


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

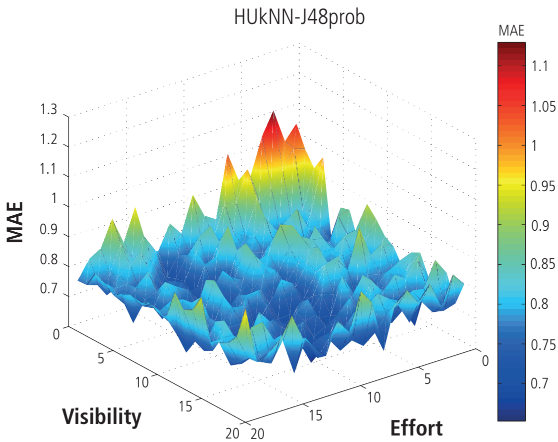


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.1: Dist. of recommendation performance of different methods regarding individuals' effort and products' visibility.

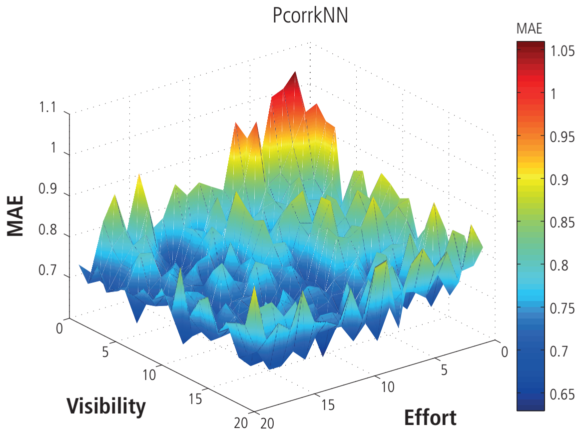


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

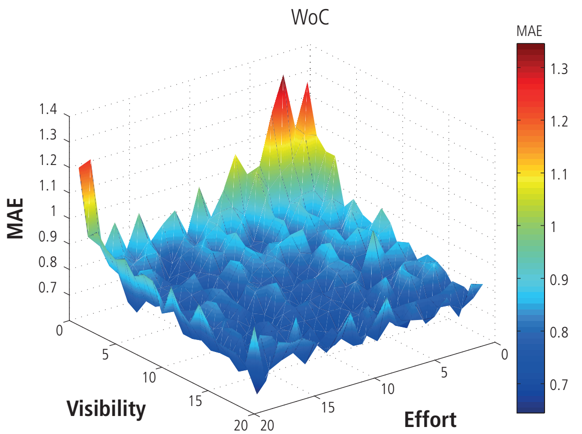


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.1: Dist. of recommendation performance of different methods regarding individuals' effort and products' visibility.

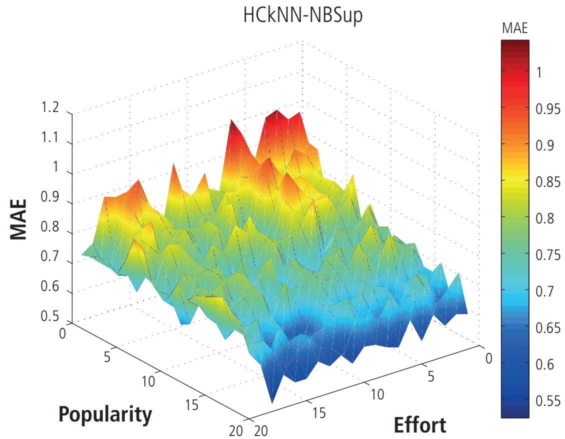


(e) Dist. of recommendation performance of PcorrkNN.

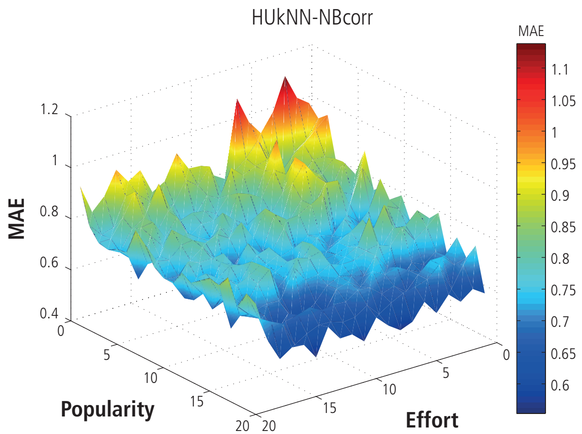


(f) Dist. of recommendation performance of WoC.

Figure D.1: Dist. of recommendation performance of different methods regarding individuals' effort and products' visibility.

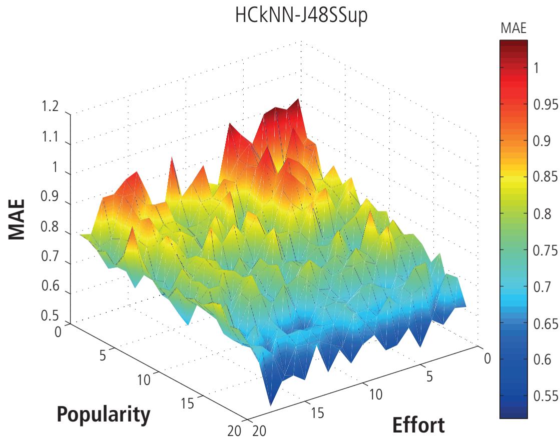


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

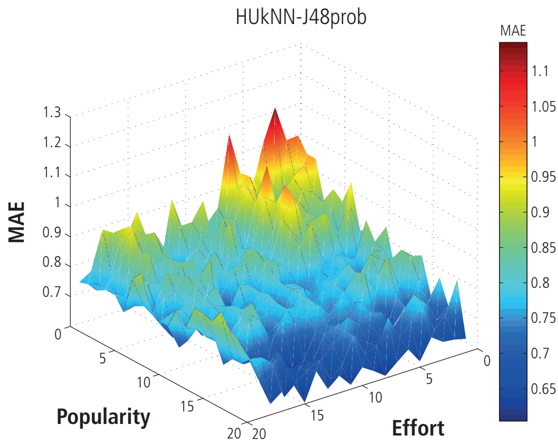


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.2: Dist. of recommendation performance of different methods regarding individuals' effort and products' popularity.

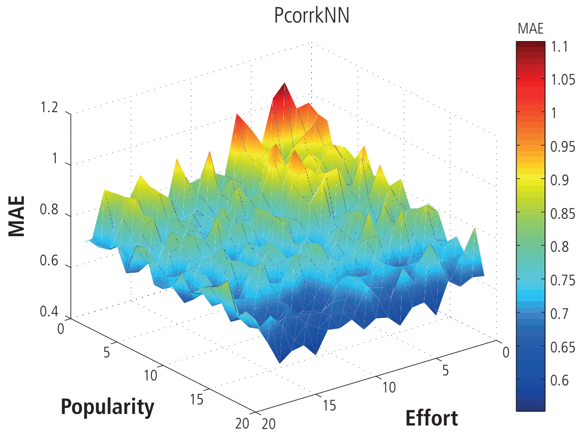


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

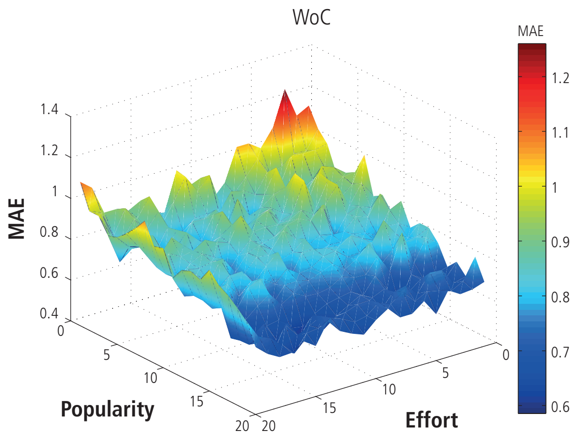


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.2: Dist. of recommendation performance of different methods regarding individuals' effort and products' popularity.

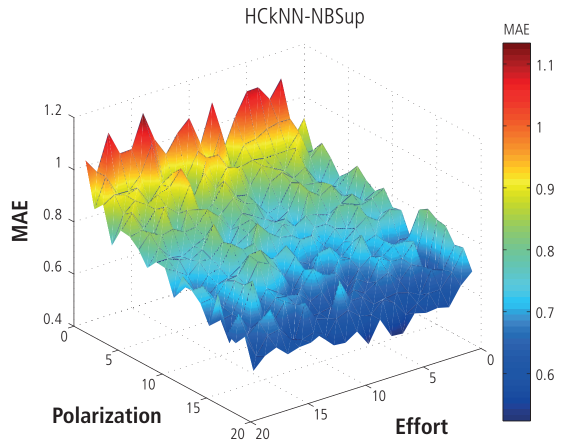


(e) Dist. of recommendation performance of PcorrkNN.

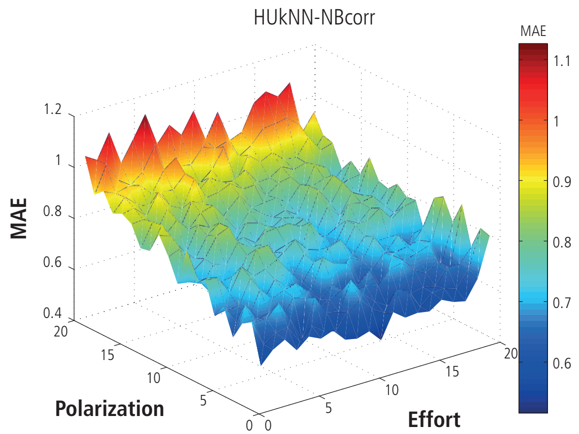


(f) Dist. of recommendation performance of WoC.

Figure D.2: Dist. of recommendation performance of different methods regarding individuals' effort and products' popularity.

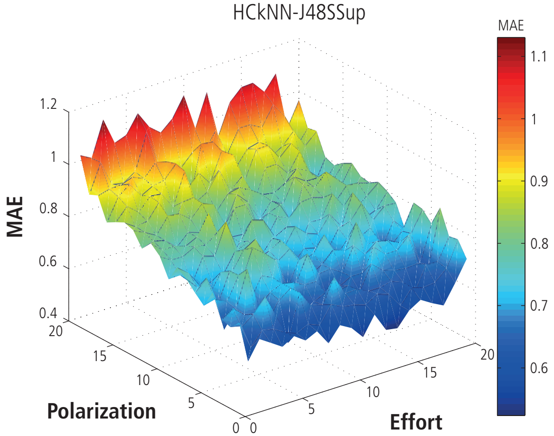


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

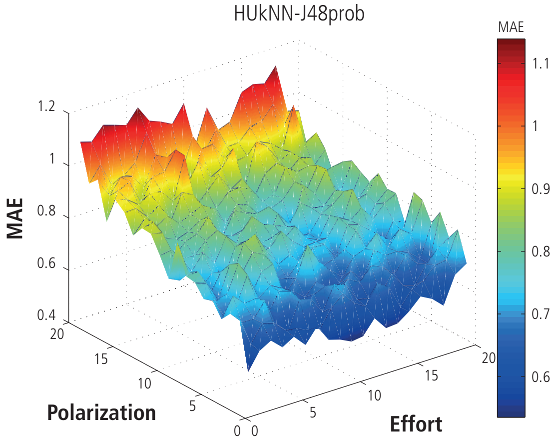


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.3: Dist. of recommendation performance of different methods regarding individuals' effort and products' polarization.

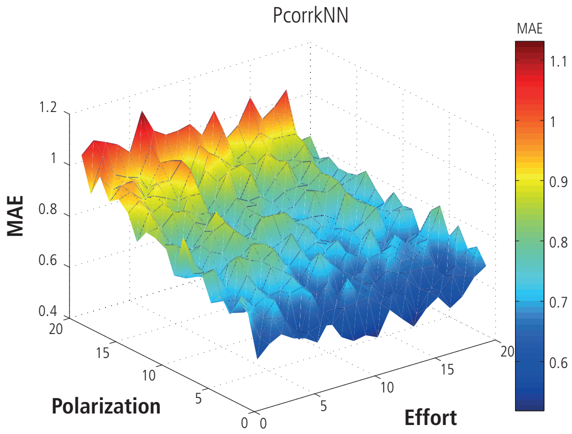


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

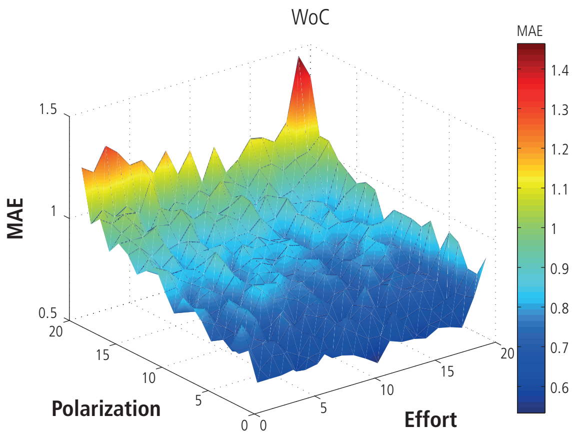


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.3: Dist. of recommendation performance of different methods regarding individuals' effort and products' polarization.

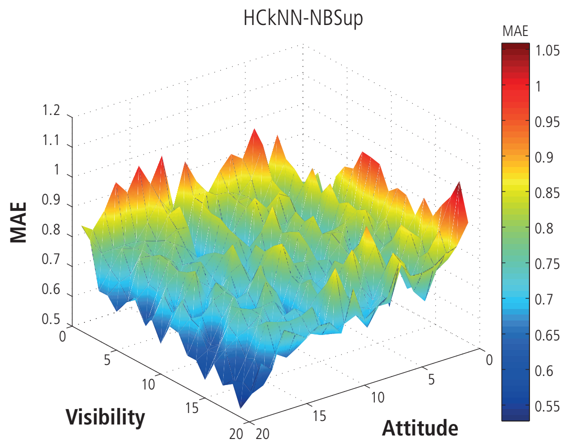


(e) Dist. of recommendation performance of PcorrkNN.

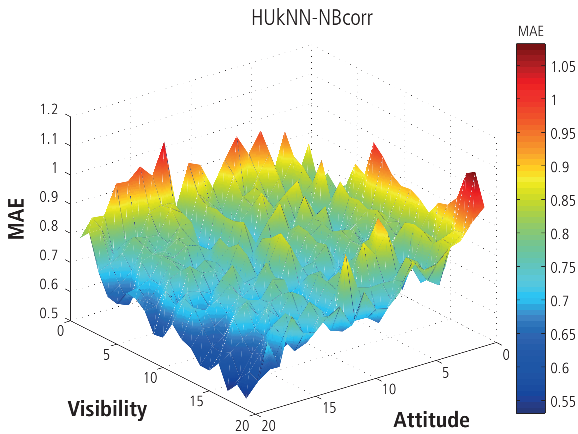


(f) Dist. of recommendation performance of WoC.

Figure D.3: Dist. of recommendation performance of different methods regarding individuals' effort and products' polarization.

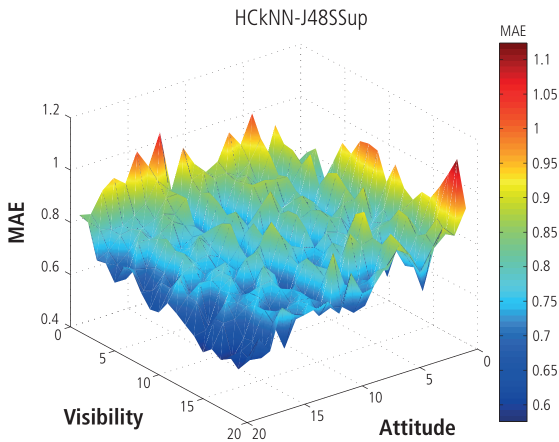


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

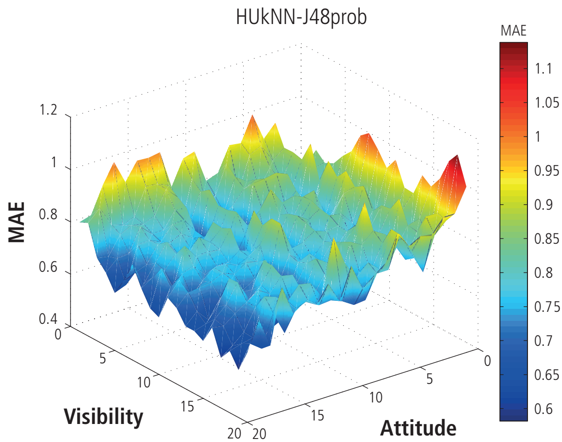


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.4: Dist. of recommendation performance of different methods regarding individuals' attitude and products' visibility.

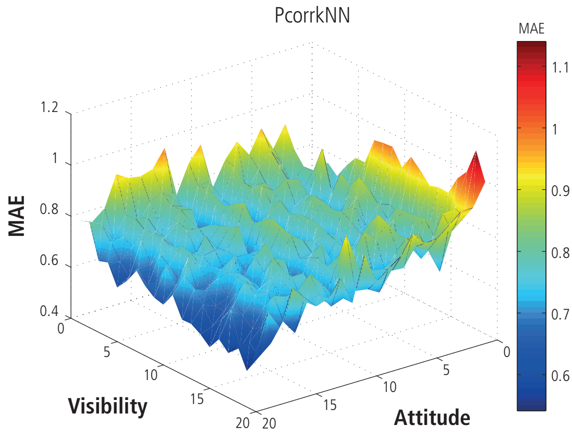


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

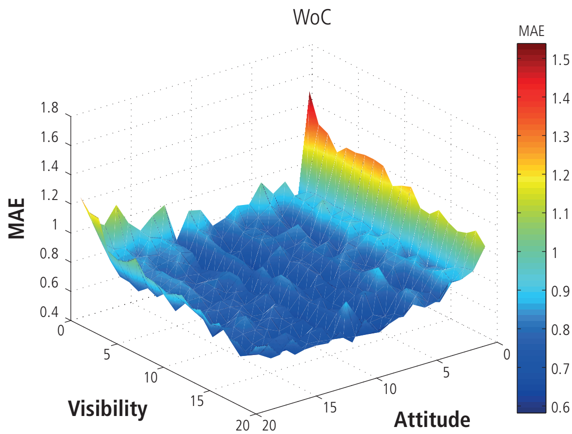


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.4: Dist. of recommendation performance of different methods regarding individuals' attitude and products' visibility.

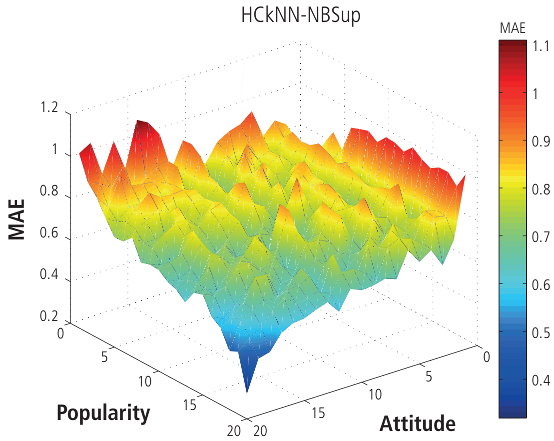


(e) Dist. of recommendation performance of PcorrkNN.

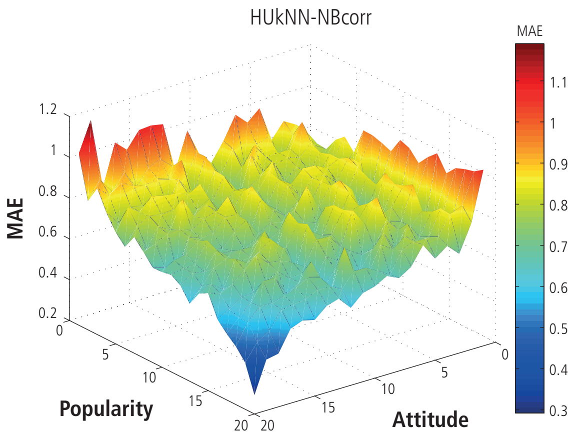


(f) Dist. of recommendation performance of WoC.

Figure D.4: Dist. of recommendation performance of different methods regarding individuals' attitude and products' visibility.

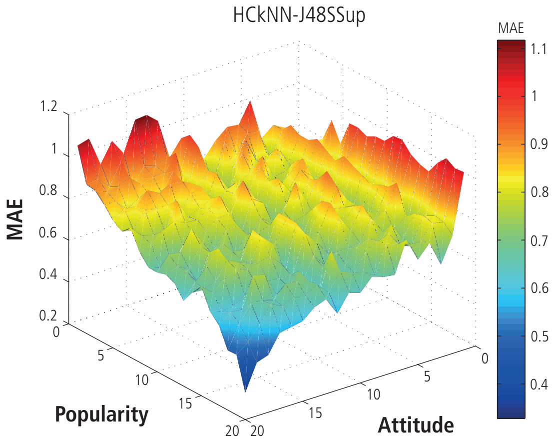


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

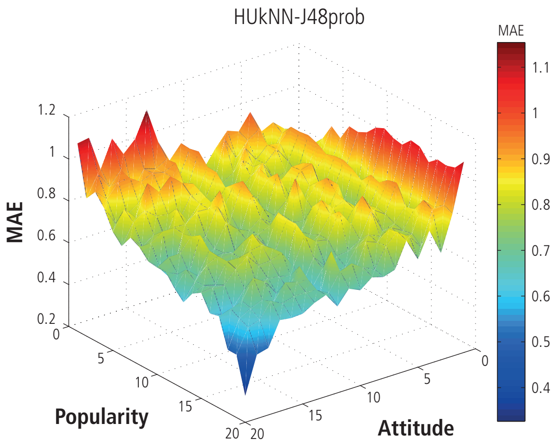


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.5: Dist. of recommendation performance of different methods regarding individuals' attitude and products' popularity.

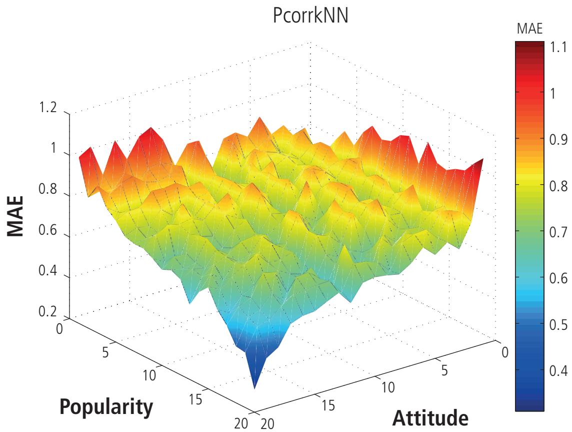


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

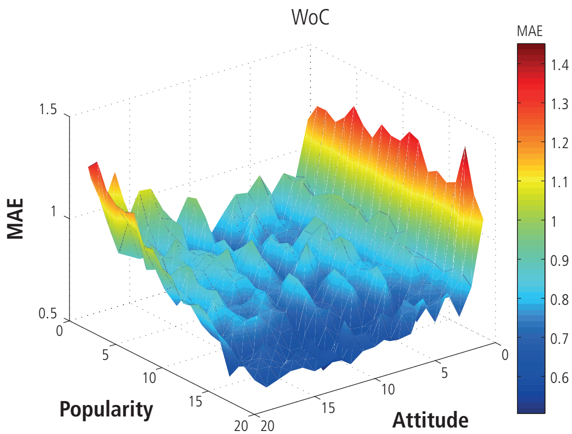


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.5: Dist. of recommendation performance of different methods regarding individuals' attitude and products' popularity.

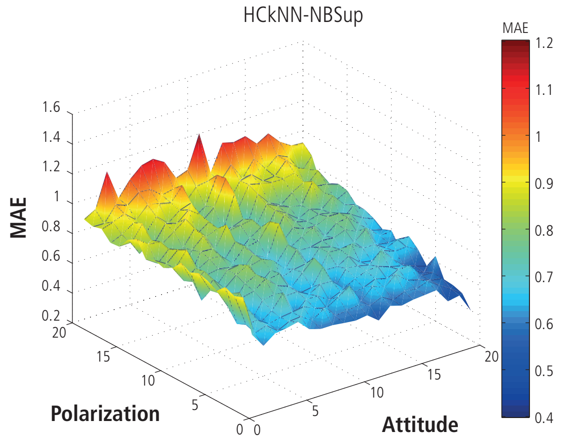


(e) Dist. of recommendation performance of PcorrkNN.

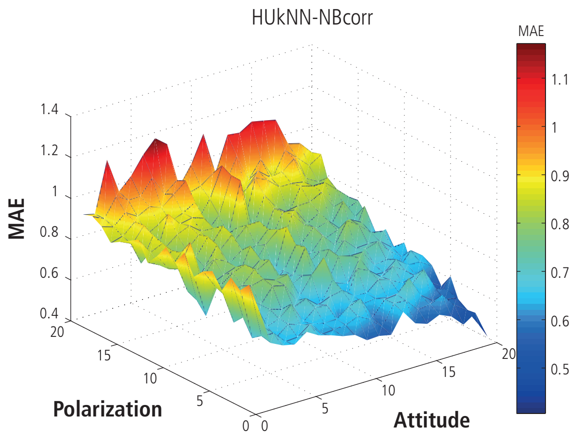


(f) Dist. of recommendation performance of WoC.

Figure D.5: Dist. of recommendation performance of different methods regarding individuals' attitude and products' popularity.

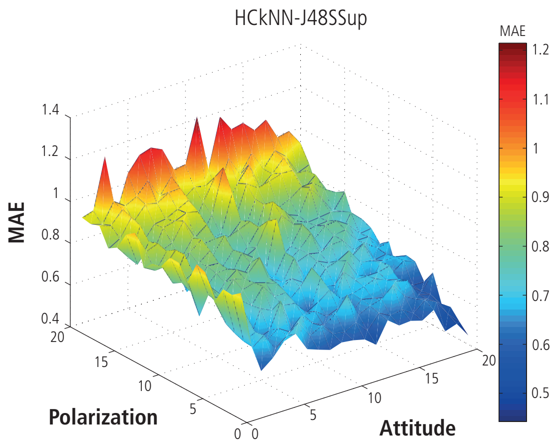


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

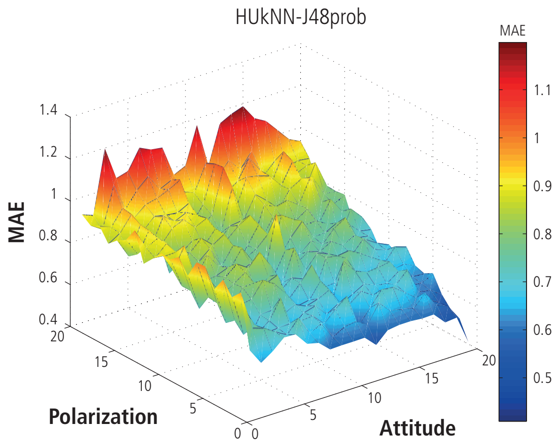


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.6: Dist. of recommendation performance of different methods regarding individuals' attitude and products' polarization.

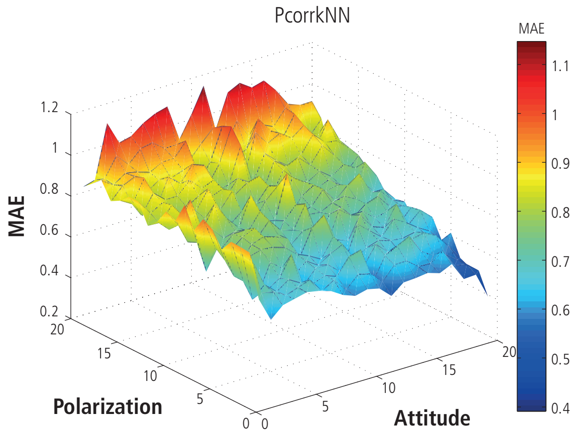


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

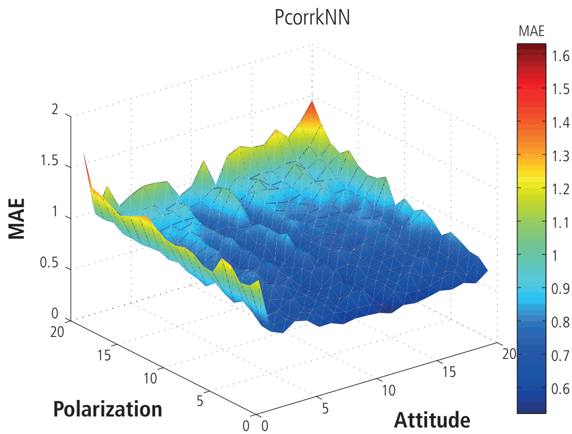


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.6: Dist. of recommendation performance of different methods regarding individuals' attitude and products' polarization.

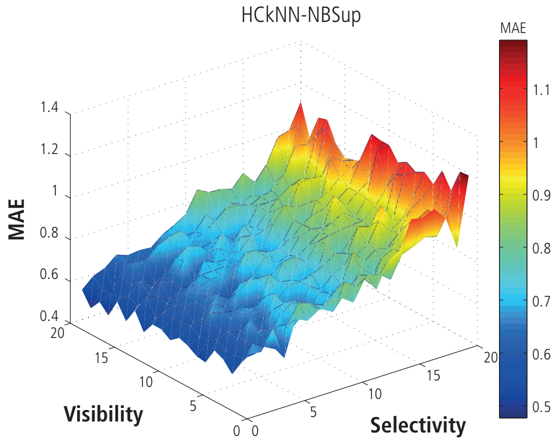


(e) Dist. of recommendation performance of PcorrkNN.

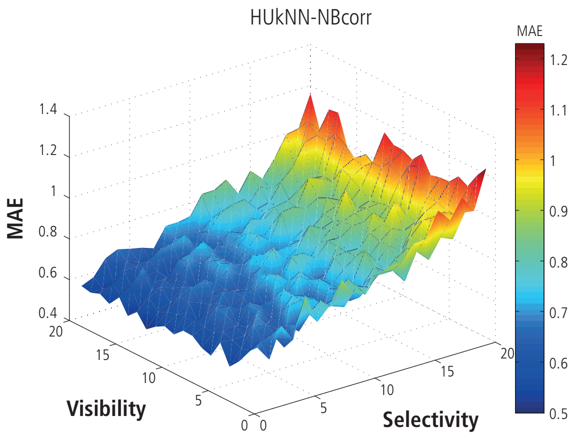


(f) Dist. of recommendation performance of WoC.

Figure D.6: Dist. of recommendation performance of different methods regarding individuals' attitude and products' polarization.

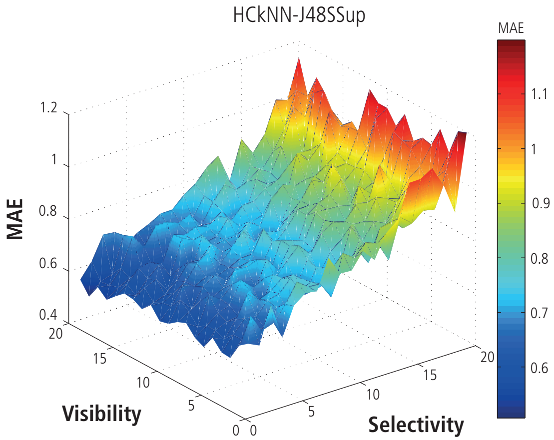


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

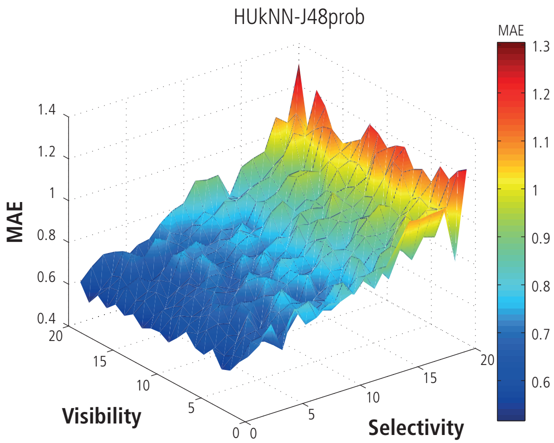


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.7: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' visibility.

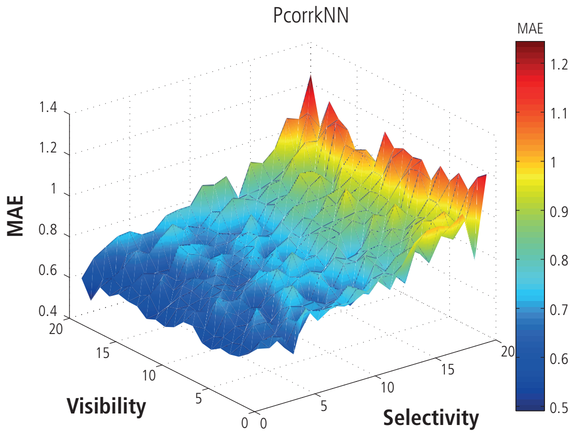


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

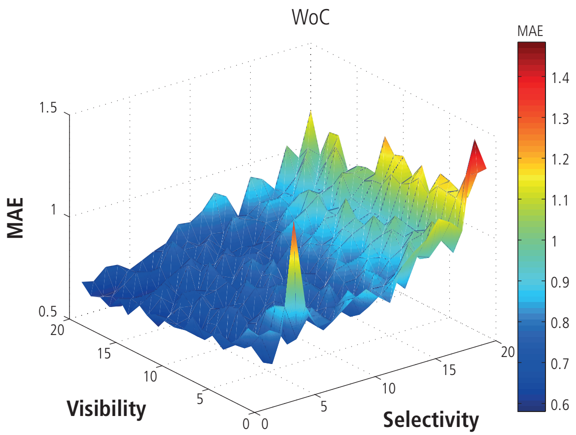


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.7: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' visibility.

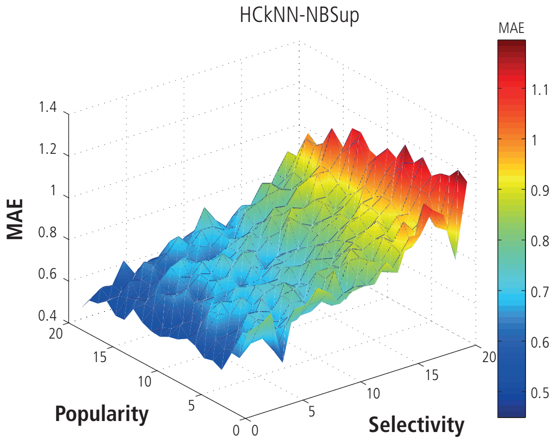


(e) Dist. of recommendation performance of PcorrkNN.

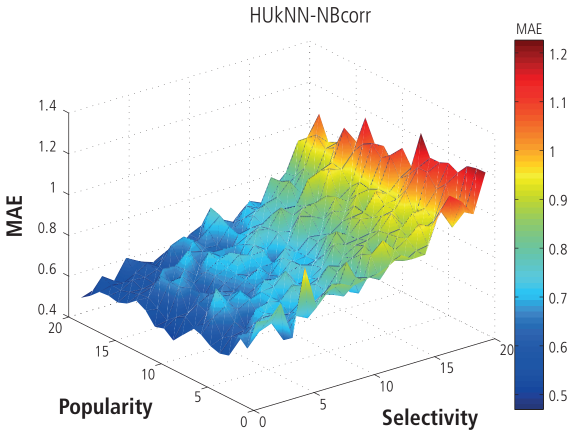


(f) Dist. of recommendation performance of WoC.

Figure D.7: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' visibility.

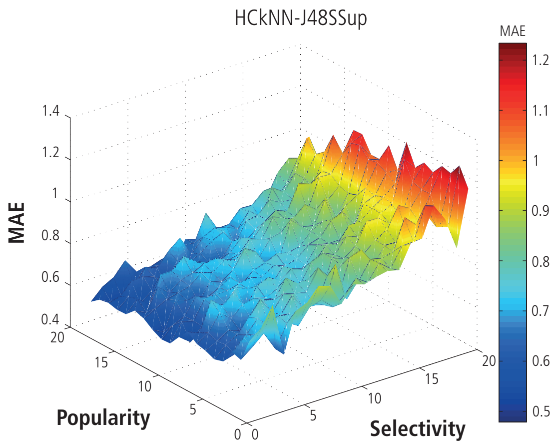


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

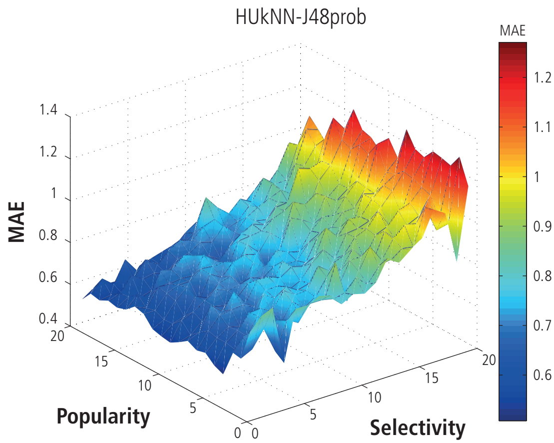


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.8: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' popularity.

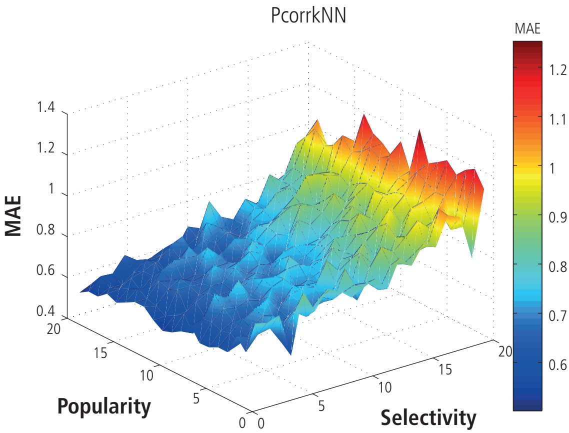


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

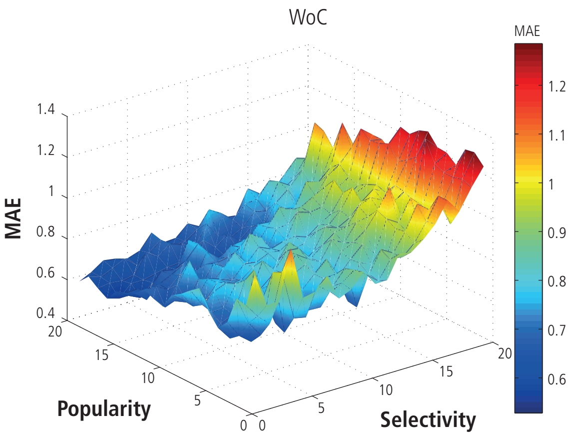


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.8: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' popularity.

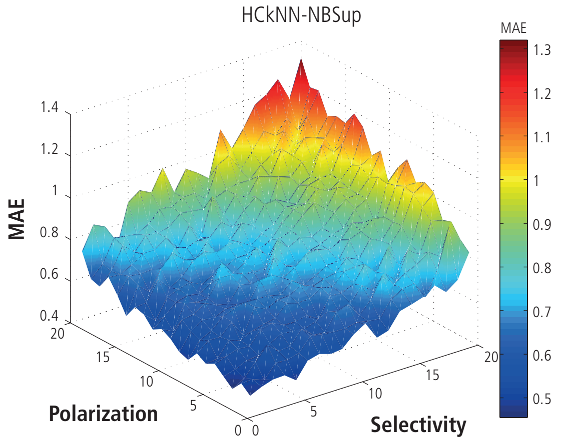


(e) Dist. of recommendation performance of PcorrkNN.

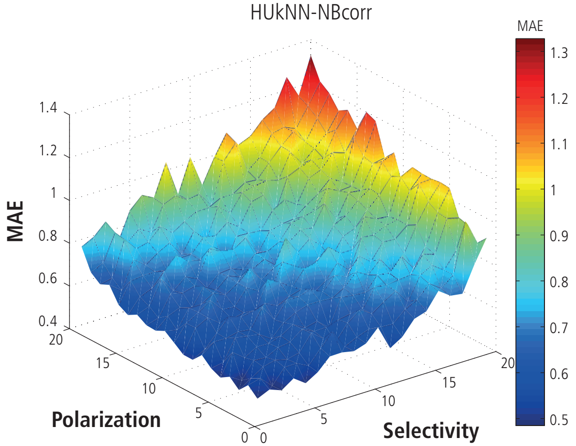


(f) Dist. of recommendation performance of WoC.

Figure D.8: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' popularity.

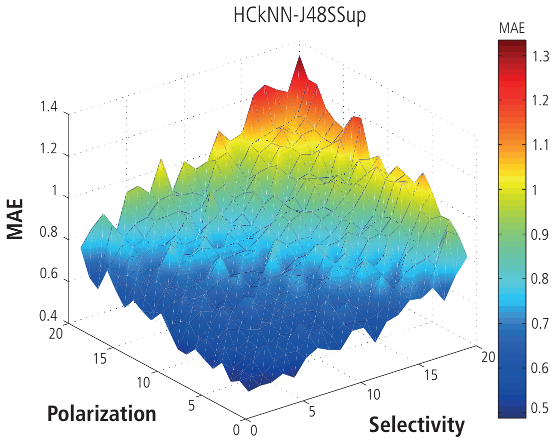


(a) Dist. of recommendation performance of *HCkNN-NBSup*.

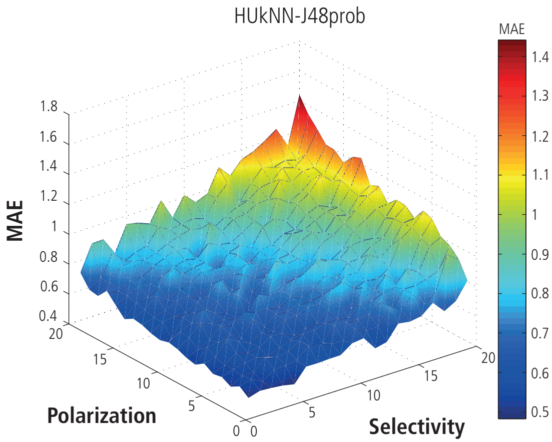


(b) Dist. of recommendation performance of *HUkNN-NBcorr*.

Figure D.9: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' polarization.

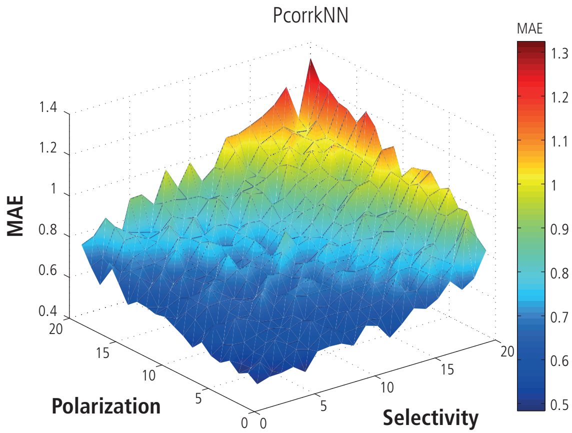


(c) Dist. of recommendation performance of *HCkNN-J48SSup*.

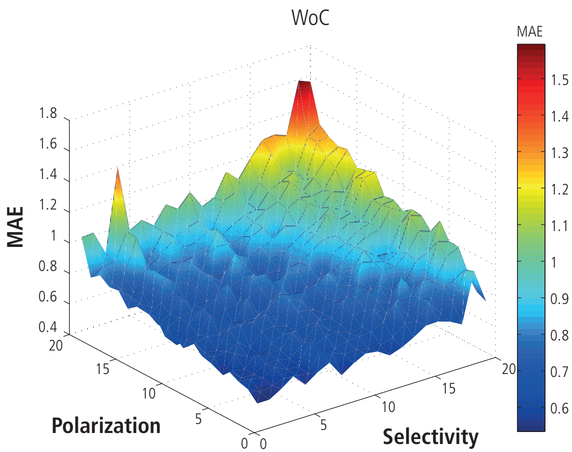


(d) Dist. of recommendation performance of *HUkNN-J48prob*.

Figure D.9: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' polarization.



(e) Dist. of recommendation performance of PcorrkNN.



(f) Dist. of recommendation performance of WoC.

Figure D.9: Dist. of recommendation performance of different methods regarding individuals' selectivity and products' polarization.

E

Comparison Between Properties and Recommendation Performance

In the following, all Pearson's correlation in all tables are significant on the significance level $\alpha = 0.01$.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.853	-0.546	0.781	-0.567	-0.236	-0.045
HUkNN-NBcorr	0.896	-0.657	0.472	-0.824	-0.606	0.191
HUkNN-SVMcorr	0.934	-0.315	0.242	-0.830	-0.560	0.044
HUkNN-J48prob	0.895	-0.375	0.881	-0.587	-0.099	0.500
HUkNN-NBprob	0.912	-0.245	0.871	-0.704	0.035	0.240
HUkNN-sJ48prob	0.890	-0.619	0.795	-0.833	-0.672	0.403
HCkNN-J48NoG	0.884	-0.232	0.113	-0.937	-0.812	0.485
HCkNN-J48Sup	0.888	-0.220	0.196	-0.939	-0.807	0.451
HCkNN-NBSup	0.934	-0.179	0.537	-0.845	-0.397	0.118
HCkNN-J48SSup	0.915	-0.117	0.257	-0.886	-0.809	0.386
HCkNN-NBSSup	0.909	-0.493	0.840	-0.231	0.098	-0.682

Table E.1: Pearson's correlations between recommendation performance difference and properties of individuals and products in the MovieLens 100k dataset.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.855	-0.571	0.715	-0.608	-0.222	-0.049
HUkNN-NBcorr	0.878	-0.551	0.298	-0.789	-0.597	0.125
HUkNN-SVMcorr	0.915	-0.285	0.132	-0.814	-0.514	-0.158
HUkNN-J48prob	0.896	-0.299	0.750	-0.596	-0.121	0.717
HUkNN-NBprob	0.839	-0.052	0.722	-0.535	-0.037	0.481
HUkNN-sJ48prob	0.933	-0.548	0.437	-0.701	-0.446	0.201
HCkNN-J48NoG	0.850	-0.295	-0.371	-0.925	-0.754	0.646
HCkNN-J48Sup	0.843	-0.212	-0.325	-0.911	-0.709	0.642
HCkNN-NBSup	0.910	-0.225	0.290	-0.873	-0.439	0.146
HCkNN-J48SSup	0.889	-0.341	-0.100	-0.888	-0.691	0.487
HCkNN-NBSSup	0.887	-0.509	0.831	-0.219	0.101	-0.696

Table E.2: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 10% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.759	-0.537	0.822	-0.625	-0.267	0.095
HUkNN-NBcorr	0.862	-0.366	0.241	-0.806	-0.555	0.176
HUkNN-SVMcorr	0.961	-0.020	0.133	-0.825	-0.597	0.234
HUkNN-J48prob	0.863	-0.161	0.831	-0.387	-0.112	0.602
HUkNN-NBprob	0.902	-0.016	0.806	-0.476	-0.012	0.369
HUkNN-sJ48prob	0.873	-0.466	-0.067	-0.888	-0.834	0.576
HCkNN-J48NoG	0.850	-0.087	-0.575	-0.812	-0.817	0.474
HCkNN-J48Sup	0.849	-0.098	-0.488	-0.911	-0.818	0.486
HCkNN-NBSup	0.904	0.097	-0.141	-0.851	-0.615	0.102
HCkNN-J48SSup	0.859	0.034	-0.372	-0.835	-0.780	0.472
HCkNN-NBSSup	0.888	-0.458	0.766	-0.143	0.117	-0.760

Table E.3: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 20% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.568	-0.619	0.683	-0.647	-0.244	-0.174
HUkNN-NBcorr	0.806	-0.531	0.180	-0.782	-0.482	0.082
HUkNN-SVMcorr	0.903	-0.378	-0.085	-0.799	-0.542	-0.022
HUkNN-J48prob	0.852	-0.163	0.624	-0.724	0.054	0.519
HUkNN-NBprob	0.884	0.033	0.619	-0.736	0.048	0.521
HUkNN-sJ48prob	0.881	-0.415	0.217	-0.818	-0.674	0.596
HCkNN-J48NoG	0.814	-0.473	-0.557	-0.931	-0.850	0.462
HCkNN-J48Sup	0.817	-0.465	-0.536	-0.928	-0.845	0.444
HCkNN-NBSup	0.858	-0.299	-0.139	-0.951	-0.583	-0.355
HCkNN-J48SSup	0.838	-0.223	-0.358	-0.951	-0.779	0.247
HCkNN-NBSSup	0.849	-0.520	0.601	-0.233	0.157	-0.795

Table E.4: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 30% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.404	-0.467	0.795	-0.614	-0.269	-0.117
HUkNN-NBcorr	0.628	-0.410	-0.067	-0.775	-0.566	0.038
HUkNN-SVMcorr	0.780	-0.139	-0.119	-0.798	-0.501	0.141
HUkNN-J48prob	0.579	0.070	0.570	-0.343	-0.013	0.540
HUkNN-NBprob	0.591	0.200	0.578	-0.343	0.156	0.490
HUkNN-sJ48prob	0.484	-0.563	0.569	-0.452	-0.507	0.549
HCkNN-J48NoG	0.074	0.209	-0.189	-0.716	-0.540	0.851
HCkNN-J48Sup	0.061	0.189	-0.097	-0.707	-0.495	0.827
HCkNN-NBSup	0.222	0.447	0.122	-0.337	0.208	0.397
HCkNN-J48SSup	0.046	0.260	0.020	-0.430	-0.405	0.708
HCkNN-NBSSup	0.675	-0.356	0.864	-0.220	0.132	-0.736

Table E.5: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 40% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.713	-0.562	0.778	-0.737	-0.248	-0.229
HUkNN-NBcorr	0.803	-0.462	0.021	-0.815	-0.486	-0.193
HUkNN-SVMcorr	0.795	-0.503	-0.150	-0.762	-0.433	-0.124
HUkNN-J48prob	0.529	-0.246	0.685	0.307	0.285	0.573
HUkNN-NBprob	0.666	-0.019	0.255	0.150	0.409	0.307
HUkNN-sJ48prob	0.567	-0.230	0.299	0.051	0.069	0.323
HCkNN-J48NoG	-0.008	0.409	-0.462	-0.339	-0.129	0.660
HCkNN-J48Sup	-0.002	0.409	-0.478	-0.251	-0.033	0.612
HCkNN-NBSup	0.238	0.421	0.183	-0.171	0.357	0.240
HCkNN-J48SSup	0.152	0.311	-0.314	-0.078	0.154	0.493
HCkNN-NBSSup	0.746	-0.377	0.871	-0.402	0.199	-0.882

Table E.6: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 50% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.636	-0.332	0.698	-0.638	-0.178	-0.384
HUkNN-NBcorr	0.585	-0.047	0.233	-0.528	-0.273	-0.472
HUkNN-SVMcorr	0.742	-0.202	-0.162	-0.595	-0.218	-0.626
HUkNN-J48prob	0.560	-0.143	0.404	0.672	0.479	0.587
HUkNN-NBprob	0.630	-0.412	0.500	0.517	0.549	0.084
HUkNN-sJ48prob	0.691	-0.039	0.726	0.455	0.365	0.302
HCkNN-J48NoG	0.098	0.466	-0.314	0.113	0.043	0.588
HCkNN-J48Sup	0.045	0.489	-0.247	0.212	0.122	0.500
HCkNN-NBSup	0.318	0.384	0.472	0.516	0.550	0.071
HCkNN-J48SSup	0.139	0.576	-0.103	0.506	0.263	0.478
HCkNN-NBSSup	0.690	-0.245	0.889	0.649	0.289	-0.792

Table E.7: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 60% sparsity degree.

ΔMAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.819	-0.416	0.708	-0.313	-0.020	-0.610
HUkNN-NBcorr	0.693	-0.559	0.495	-0.241	-0.303	0.028
HUkNN-SVMcorr	0.750	-0.311	-0.332	-0.470	-0.320	-0.447
HUkNN-J48prob	0.531	0.179	0.221	0.816	0.368	0.522
HUkNN-NBprob	0.506	0.391	0.206	0.764	0.446	0.457
HUkNN-sJ48prob	0.745	-0.369	0.314	0.552	0.194	0.479
HCkNN-J48NoG	0.395	0.305	-0.090	0.408	0.178	0.662
HCkNN-J48Sup	0.303	0.316	-0.067	0.470	0.214	0.668
HCkNN-NBSup	0.483	0.607	0.187	0.721	0.450	0.337
HCkNN-J48SSup	0.462	0.334	-0.070	0.749	0.315	0.655
HCkNN-NBSSup	0.712	-0.154	0.845	0.790	0.404	-0.868

Table E.8: Pearson's correlations between recommendation performance difference and properties of individuals and products in the dataset with 70% sparsity degree.

∆MAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	0.515	-0.132	0.644	0.034	0.143	-0.360
HUkNN-NBcorr	0.365	-0.150	0.133	0.029	0.187	0.079
HUkNN-SVMcorr	0.684	-0.140	0.076	-0.304	0.091	-0.289
HUkNN-J48prob	0.204	0.285	0.122	0.475	0.446	0.643
HUkNN-NBprob	0.304	0.386	-0.003	0.534	0.483	0.538
HUkNN-sJ48prob	0.511	-0.221	0.256	0.327	0.315	0.620
HCkNN-J48NoG	0.281	0.143	-0.425	0.405	0.253	0.657
HCkNN-J48Sup	0.252	0.102	-0.418	0.437	0.279	0.641
HCkNN-NBSup	0.236	0.504	-0.039	0.616	0.568	0.449
HCkNN-J48SSup	0.308	0.296	-0.361	0.506	0.381	0.703
HCkNN-NBSSup	0.413	-0.009	0.788	0.884	0.580	-0.743

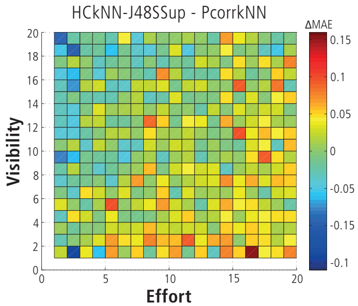
Table E.9: Pearson’s correlations between recommendation performance difference and properties of individuals and products in the dataset with 80% sparsity degree.

∆MAE PcorrkNN	Effor.	Attit.	Selec.	Visib.	Popul.	Polar.
HUkNN-J48corr	-0.150	-0.222	-0.284	-0.868	-0.745	-0.150
HUkNN-NBcorr	-0.434	0.207	-0.581	-0.875	-0.685	-0.434
HUkNN-SVMcorr	0.124	-0.159	-0.597	-0.919	-0.710	-0.124
HUkNN-J48prob	-0.668	0.359	-0.546	-0.736	-0.260	-0.668
HUkNN-NBprob	-0.336	0.446	-0.655	-0.773	-0.206	-0.336
HUkNN-sJ48prob	-0.447	0.393	-0.589	-0.725	-0.354	-0.447
HCkNN-J48NoG	-0.216	0.242	-0.641	-0.766	-0.446	-0.216
HCkNN-J48Sup	-0.226	0.246	-0.642	-0.756	-0.440	-0.226
HCkNN-NBSup	-0.502	0.470	-0.638	-0.688	-0.163	-0.502
HCkNN-J48SSup	-0.542	0.345	-0.683	-0.798	-0.265	-0.542
HCkNN-NBSSup	-0.569	0.389	0.450	0.344	0.427	-0.569

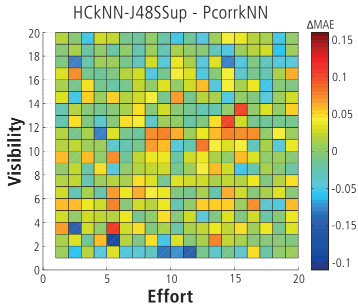
Table E.10: Pearson’s correlations between recommendation performance difference and properties of individuals and products in the dataset with 90% sparsity degree.

F

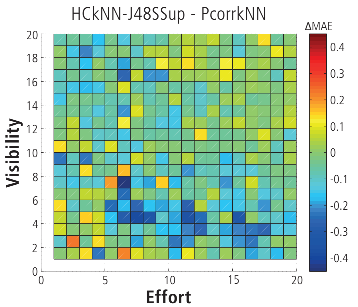
Comparison Between Recomm. Perform. regarding Cold-Start Behavior



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

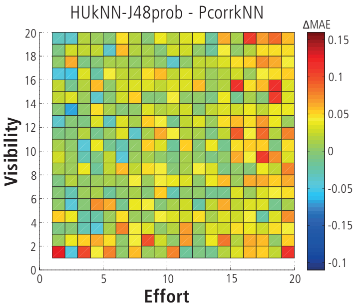


(b) Comparison of recommendation performance for the dataset 40% sparsity.

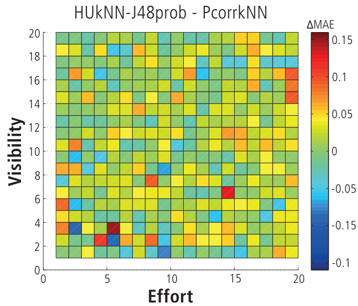


(c) Comparison of recommendation performance for the dataset 40% sparsity.

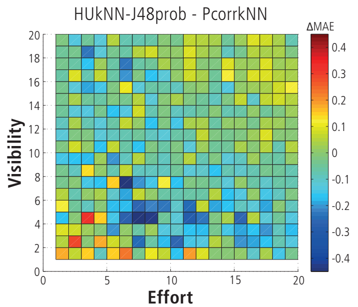
Figure F.1: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' effort and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

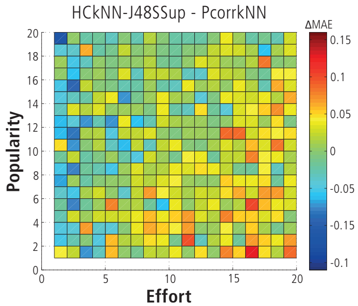


(b) Comparison of recommendation performance for the dataset 40% sparsity.

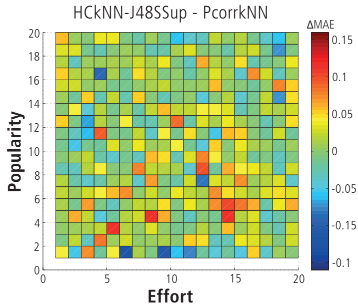


(c) Comparison of recommendation performance for the dataset 40% sparsity.

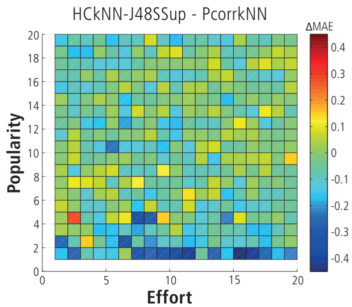
Figure F.2: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' effort and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

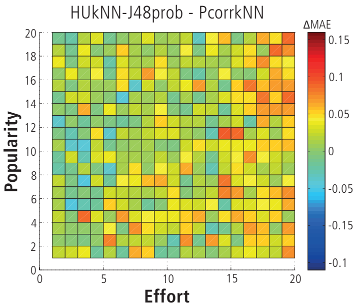


(b) Comparison of recommendation performance for the dataset 40% sparsity.

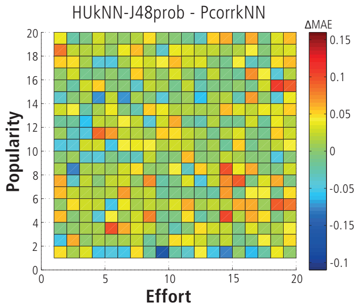


(c) Comparison of recommendation performance for the dataset 40% sparsity.

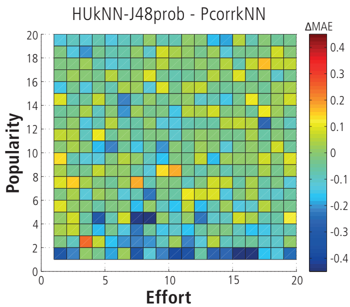
Figure F.3: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' effort and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

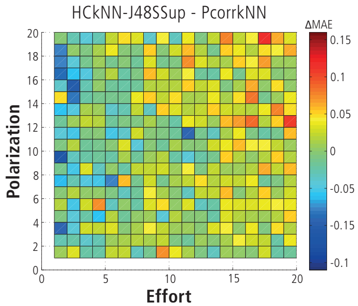


(b) Comparison of recommendation performance for the dataset 40% sparsity.

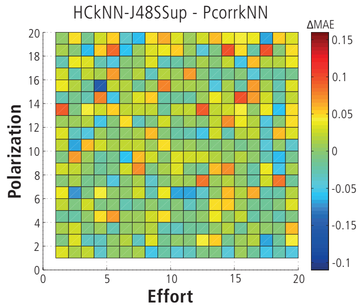


(c) Comparison of recommendation performance for the dataset 40% sparsity.

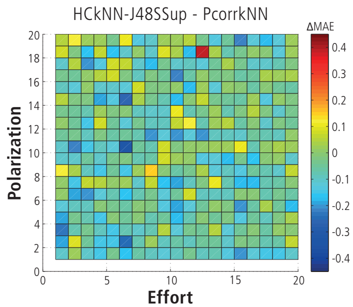
Figure F.4: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' effort and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

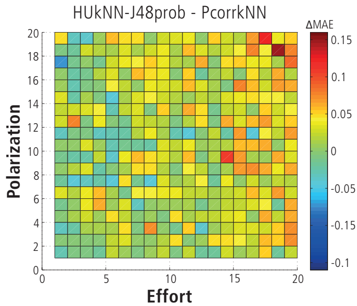


(b) Comparison of recommendation performance for the dataset 40% sparsity.

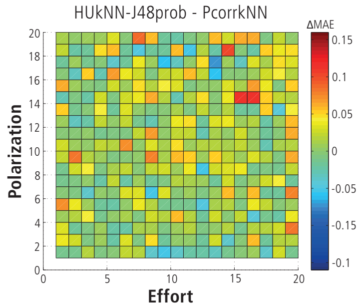


(c) Comparison of recommendation performance for the dataset 40% sparsity.

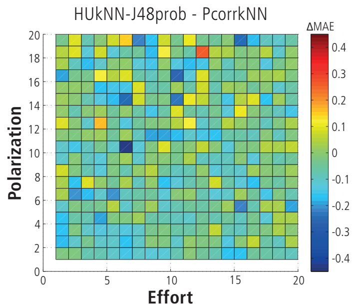
Figure F.5: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' effort and products' polarization.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

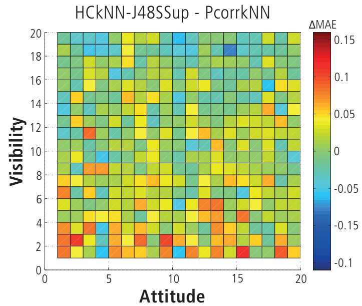


(b) Comparison of recommendation performance for the dataset 40% sparsity.

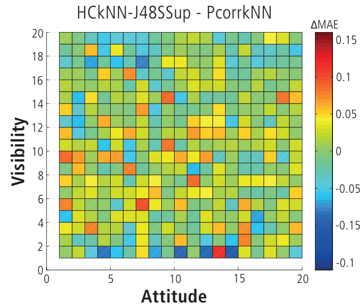


(c) Comparison of recommendation performance for the dataset 40% sparsity.

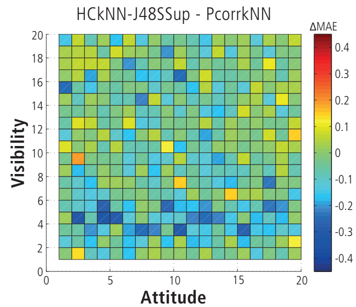
Figure F.6: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' effort and products' polarization.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

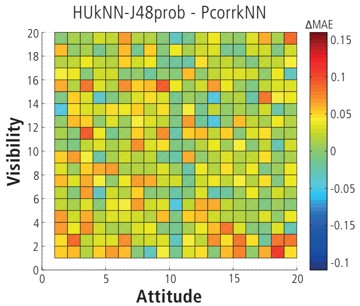


(b) Comparison of recommendation performance for the dataset 40% sparsity.

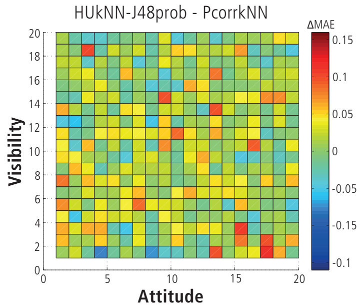


(c) Comparison of recommendation performance for the dataset 40% sparsity.

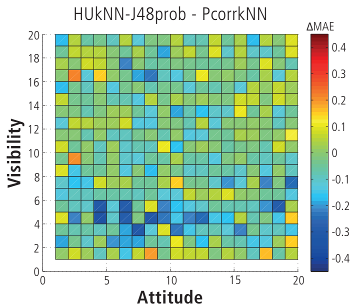
Figure F.7: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' attitude and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

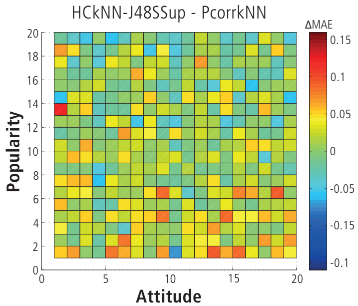


(b) Comparison of recommendation performance for the dataset 40% sparsity.

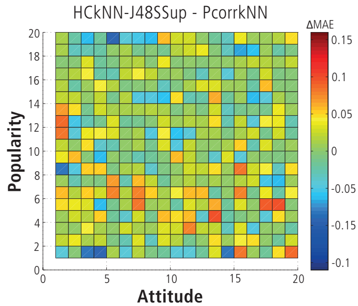


(c) Comparison of recommendation performance for the dataset 40% sparsity.

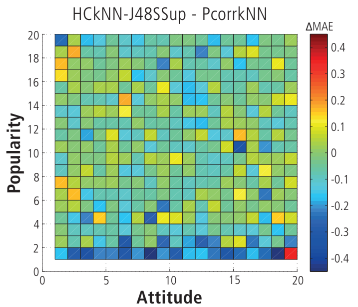
Figure F.8: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' attitude and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

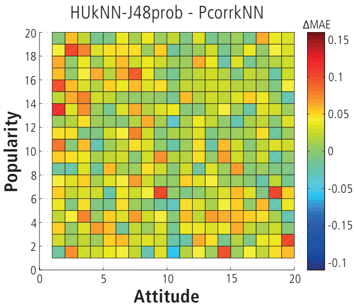


(b) Comparison of recommendation performance for the dataset 40% sparsity.

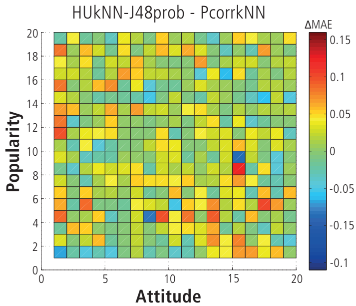


(c) Comparison of recommendation performance for the dataset 40% sparsity.

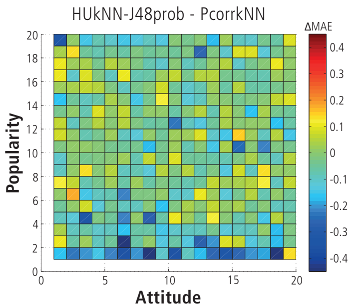
Figure F.9: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' attitude and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

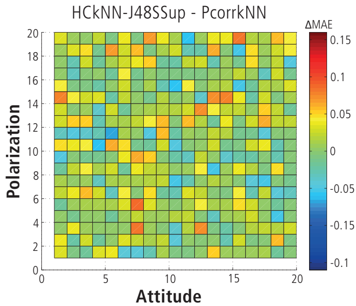


(b) Comparison of recommendation performance for the dataset 40% sparsity.

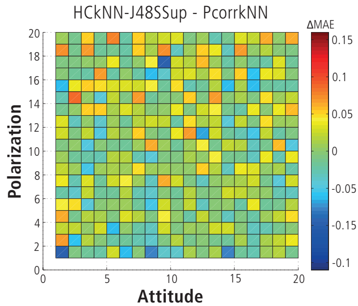


(c) Comparison of recommendation performance for the dataset 40% sparsity.

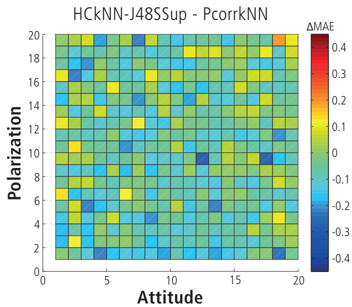
Figure F.10: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' attitude and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

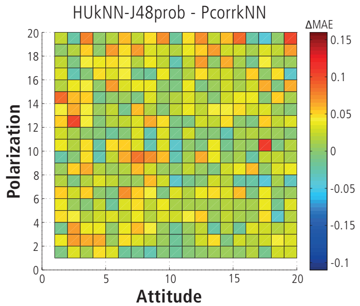


(b) Comparison of recommendation performance for the dataset 40% sparsity.

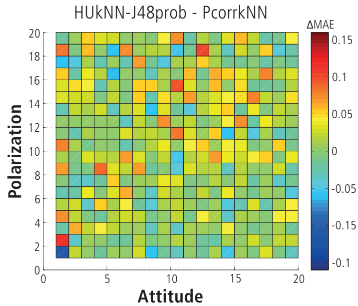


(c) Comparison of recommendation performance for the dataset 40% sparsity.

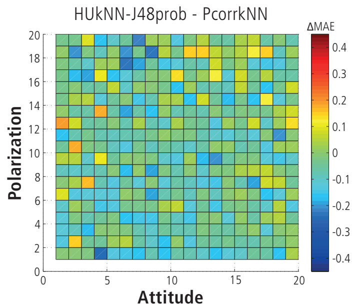
Figure F.11: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' attitude and products' polarization.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

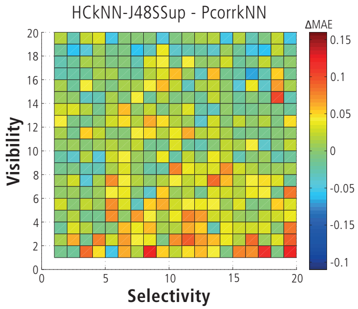


(b) Comparison of recommendation performance for the dataset 40% sparsity.

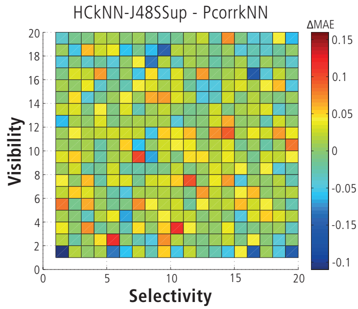


(c) Comparison of recommendation performance for the dataset 40% sparsity.

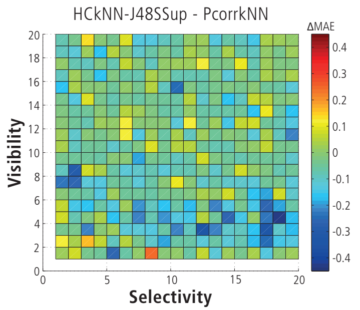
Figure F.12: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' attitude and products' polarization.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

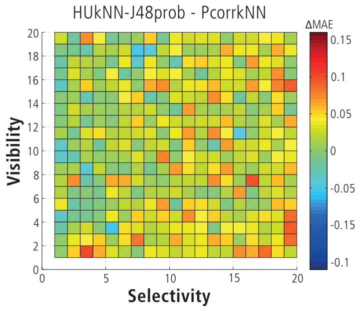


(b) Comparison of recommendation performance for the dataset 40% sparsity.

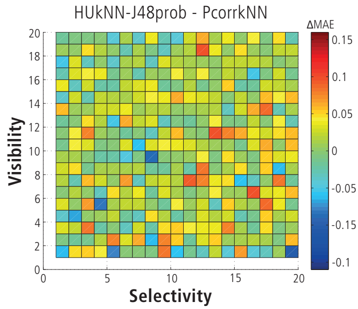


(c) Comparison of recommendation performance for the dataset 40% sparsity.

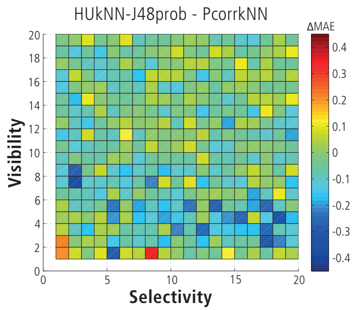
Figure F.13: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' selectivity and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

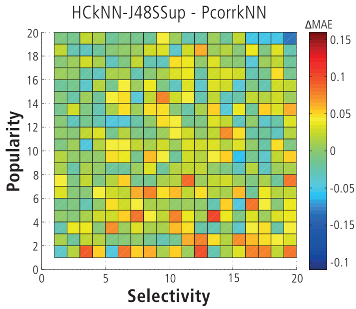


(b) Comparison of recommendation performance for the dataset 40% sparsity.

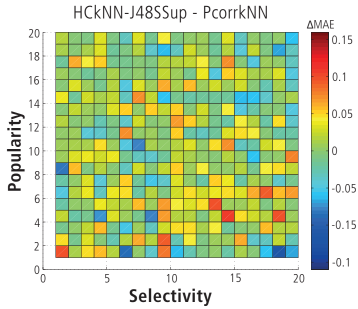


(c) Comparison of recommendation performance for the dataset 40% sparsity.

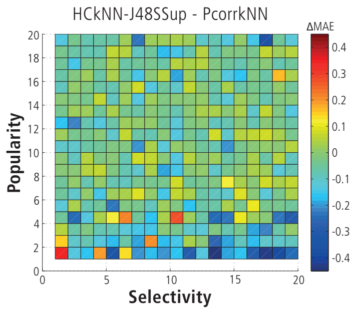
Figure F.14: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' selectivity and products' visibility.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

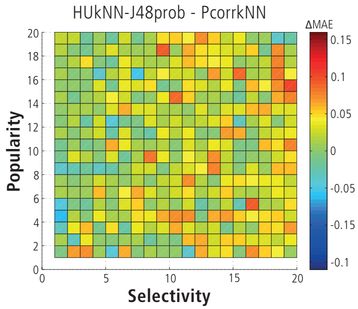


(b) Comparison of recommendation performance for the dataset 40% sparsity.

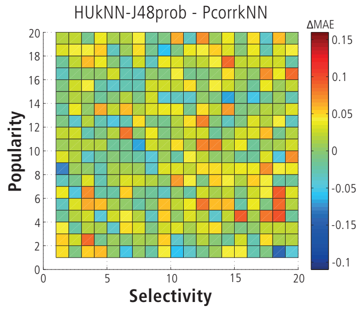


(c) Comparison of recommendation performance for the dataset 40% sparsity.

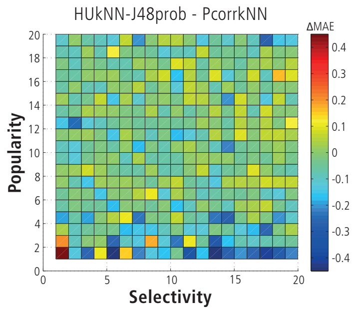
Figure F.15: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' selectivity and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

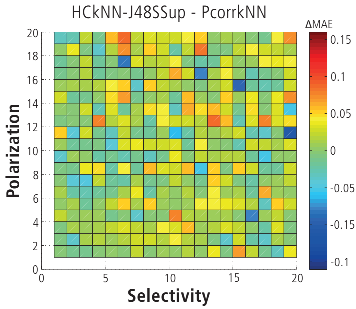


(b) Comparison of recommendation performance for the dataset 40% sparsity.

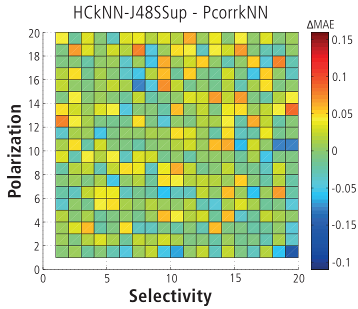


(c) Comparison of recommendation performance for the dataset 40% sparsity.

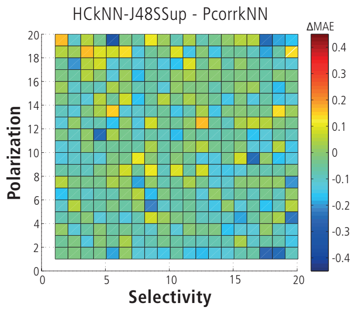
Figure F.16: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' selectivity and products' popularity.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).

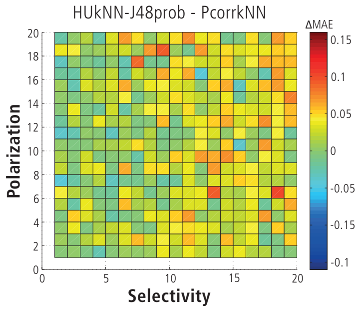


(b) Comparison of recommendation performance for the dataset 40% sparsity.

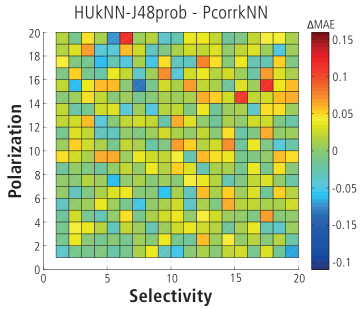


(c) Comparison of recommendation performance for the dataset 40% sparsity.

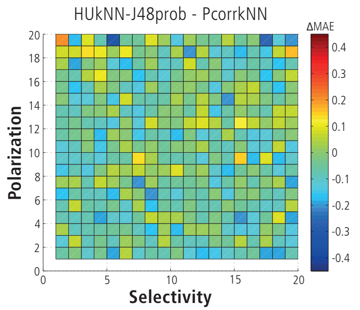
Figure F.17: Comparison of recommendation performance and cold-start behavior of *HCkNN-J48SSup* and PcorrkNN regarding individuals' selectivity and products' polarization.



(a) Comparison of recommendation performance for the dataset 0% sparsity (i.e., original MovieLens 100k dataset).



(b) Comparison of recommendation performance for the dataset 40% sparsity.



(c) Comparison of recommendation performance for the dataset 40% sparsity.

Figure F.18: Comparison of recommendation performance and cold-start behavior of *HUkNN-J48prob* and PcorrkNN regarding individuals' selectivity and products' polarization.

G

Publications

- Amancio Bouza, Abraham Bernstein, (Partial) User Preference Similarity as Classification-Based Model Similarity, *Inductive Reasoning and Machine Learning for the Semantic Web [Special Issue], Semantic Web Journal*, (accepted with major revision)

- Amancio Bouza, Gerald Reif, Abraham Bernstein, Probabilistic Partial User Model Similarity for Collaborative Filtering, In *Proceedings of the 1st International Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS 2009) at the 6th European Semantic Web Conference (ESWC 2009)*, June 2009.

- Tobias Bannwart, Amancio Bouza, Gerald Reif, Abraham Bernstein, Private Cross-page Movie Recommendations with the Firefox add-on OMORE, *8th International Semantic Web Conference (ISWC 2009)*, October 2009.

- Amancio Bouza, Gerald Reif, Abraham Bernstein, Harald C. Gall, SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems, In *Proceedings of the 7th International Semantic Web Conference*, October 2008. (Poster)

Bibliography

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–74.

[Anand et al., 2007] Anand, S. S., Kearney, P., and Shapcott, M. (2007). Generating semantically enriched user profiles for web personalization. *ACM Transactions on Internet Technology*, 7(4).

[Anderson, 2006] Anderson, C. (2006). *The Long Tail. Why the Future of Business Is Selling Less of More*. Hyperion, 1st edition.

[Balabanovic and Shoham, 1997] Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. In *Communications of the ACM*, volume 40, pages 66–72, New York, NY, USA. ACM.

[Bannwart et al., 2009] Bannwart, T., Bouza, A., Reif, G., and Bernstein, A. (2009). Private cross-page movie recommendations with the firefox add-on omore. In *Semantic Web Challenge at the 8th International Semantic Web Conference*.

[Basu et al., 1998] Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 714–720. AAAI Press.

[Bell and Koren, 2007] Bell, R. M. and Koren, Y. (2007). Improved neighborhood-based collaborative filtering. In *The 13th International*

Conference on Knowledge Discovery and Data Mining in KDD Cup, San Jose, California.

[Berkovsky et al., 2007] Berkovsky, S., Kuflik, T., and Ricci, F. (2007). Mediation of user models for enhanced personalization in recommender systems. In *User Modeling and User-Adapted Interaction*, volume 18, pages 245–286.

[Bernanke and Frank, 2007] Bernanke, B. S. and Frank, R. H. (2007). *Principles of Micro Economics*. McGraq-Hill/Irwin.

[Bernstein et al., 2005] Bernstein, A., Kaufmann, E., Kiefer, C., and Bürki, C. (2005). SimPack: A Generic Java Library for Similiarity Measures in Ontologies. Technical report, Department of Informatics, University of Zurich.

[Bizer and Cyganiak, 2006] Bizer, C. and Cyganiak, R. (2006). D2r server - publishing releational databases on the semantic web. In *5th International Semantic Web Conference.*

[Blume and Easley., 2008] Blume, L. E. and Easley., D. (2008). Rationality. In Durlauf, S. N. and Blume, L. E., editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, Basingstoke.

[Boatwright and Nunes, 2001] Boatwright, P. and Nunes, J. C. (2001). Reducing assortment: An attribute-based approach. *The Journal of Marketing*, 65(3):50–63.

[Bouza and Bernstein, 2012] Bouza, A. and Bernstein, A. (2012). Partial user preference similarity as classification-based model similarity. *Semantic Web Journal (Special Issue)*.

[Bouza et al., 2009] Bouza, A., Reif, G., and Bernstein, A. (2009). Probabilistic partial user model similarity for collaborative filtering. In *Proceedings* *of the 1st International Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS2009) at the 6th European Semantic Web Conference (ESWC2009)*, volume 474 of *ISSN 1613-0073*.

[Bouza et al., 2008] Bouza, A., Reif, G., Bernstein, A., and Gall, H. (2008). Semtree: Ontology-based decision tree algorithm for recommender systems. In *7th International Semantic Web Conference (ISWC)*.

[Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *14th Conference on Uncertainty in AI*, pages 43–52. Morgan Kaufmann.

[Brynjolfsson and Smith, 2000] Brynjolfsson, E. and Smith, M. D. (2000). Frictionless commerce? a comparison of internet and conventional retailers. *Management Science*, 46(46):563–585.

[Brynjolfsson et al., 2003] Brynjolfsson, E., Smith, M. D., and Hu, Y. J. (2003). Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science*, 49(11).

[Burke, 2002] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

[Cantador et al., 2008] Cantador, I., Bellogín, A., and Castells, P. (2008). A multilayer ontology-based hybrid recommendation model. *AI Communcations*, 21(2–3):202–210.

[Cantador and Castells, 2006] Cantador, I. and Castells, P. (2006). Building emergent social networks and group profiles by semantic user preference clustering. In *2nd International Workshop on Semantic Network Analysis (SNA 2006), at the 3rd European Semantic Web Conference (ESWC 2006)*, pages 40–53, Budva, Motenegro.

[Chajewska and Koller, 2000] Chajewska, U. and Koller, D. (2000). Utilities as random variables: Density estimation and structure discovery. In *Uncertainty in Artificial Intelligence*, pages 63–71.

[Corbin and Strauss, 2008] Corbin, J. and Strauss, A. (2008). *Basics of qualitative research : techniques and procedures for developing grounded theory*. SAGE, Los Angeles, CA, USA, 3rd edition.

[Dean and Schreiber, 2004] Dean, M. and Schreiber, G. (2004). *OWL Web Ontology Language Reference*. W3C Recommendation. `http://www.w3.org/TR/owl-ref/`.

[Dhar, 1997] Dhar, R. (1997). Consumer preference for a no-choice option. *Journal of Consumer Research*, 24(2):215–231.

[Dixit and Stiglitz, 1977] Dixit, A. K. and Stiglitz, J. E. (1977). Monopolistic competition and optimum product diversity. *The American Economic Review*, 67(3):297–308.

[Ge et al., 2010] Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, number 4 in RecSys '10, pages 257–260, New York, NY, USA. ACM.

[Glaser and Strauss, 1967] Glaser, B. G. and Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, New York, NY, USA.

[Golbeck et al., 2003] Golbeck, J., Parsia, B., and Hendler, J. (2003). Trust networks on the semantic web. In *Proceedings of Cooperative Information Agents 2003*, pages 27–29, Helsinki, Finland.

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70.

[Good et al., 1999] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. In *AAAI '99/IAAI '99: Proceedings of the 16th national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, number 16, pages 439–446, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[Gourville and Soman, 2005] Gourville, J. T. and Soman, D. (2005). Overchoice and assortment type: When and why variety backfires. *Marketing Science*, 24(3):382–395.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Journal Knowledge Acquisition*, 2(5):199–220.

[Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *IGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA. ACM.

[Herlocker et al., 2004] Herlocker, J. L., Konstan, J. A., Reveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53.

[Hinz et al., 2011] Hinz, O., Eckert, J., and Skiera, B. (2011). Drivers of the long tail phenomenon: An empirical analysis. *Journal of Management Information Systems*, 27(4):43 – 69.

[Hoch et al., 1999] Hoch, S. J., Bradlow, E. T., and Wansink, B. (1999). The variety of an assortment. *Marketing Science*, 18(4):527–546.

[Kahn, 1995] Kahn, B. E. (1995). Consumer variety-seeking among goods and services: An integrative review. *Journal of Retailing and Consumer Services*, 2(3):139–148.

[Kautz et al., 1997] Kautz, H., Selman, B., and Shah, M. (1997). Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65.

[Konstan, 2004] Konstan, J. A. (2004). Introduction to recommender systems: Algorithms and evaluation. *ACM Transactions on Information Systems*, 22(1):1–4.

[Kudenko, 2000] Kudenko, D. (2000). Ontology-based constructive induction (extended abstract).

[Kurzynski, 1983] Kurzynski, M. W. (1983). The optimal strategy of a tree classifier. *Pattern Recognition*, 16:81–87.

[Lawrence and Urtasun, 2009] Lawrence, N. D. and Urtasun, R. (2009). Non-linear matrix factorization with gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 601–608, New York, NY, USA. ACM.

[Melville et al., 2002] Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *18th national conference on Artificial intelligence*, pages 187–192. American Association for Artificial Intelligence.

[Middleton et al., 2002] Middleton, S. E., Alani, H., and de Roure, D. C. (2002). Exploiting synergy between ontologies and recommender systems. In *International Workshop on the Semantic Web, Proceedings of the 11th International World Wide Web Conference*, Hawaii.

[Middleton et al., 2004] Middleton, S. E., Shadbolt, N. R., and de Roure, D. C. (2004). Ontological user profiling in recommender systems. In *ACM Transactions on Information Systems*, volume 22, pages 54–88.

[Mitchel, 1997] Mitchel, T. M. (1997). *Machine Learning*. McGraw Hill Higher Education.

[Noy and McGuiness, 2001] Noy, N. F. and McGuiness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems - Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition edition.

[Pirró and Euzenat, 2010] Pirró, G. and Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. In *Proceedings of the 9th International Semantic Web Conference on the Semantic Web*, pages 615–630, Berlin, Heidelberg. Spriner-Verlag.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

[Rashid et al., 2006] Rashid, A. M., Lam, S. K., Karypis, G., and Riedl, J. (2006). Clustknn: A highly scalable hybrid model- and memory-based cg algorithm. In *WEBKDD*.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA. ACM.

[Resnick et al., 1997] Resnick, P., Varian, H. R., and Editors, G. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

[Safavian and Landgrebe, 1991] Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. In *IEEE Transactions on Systems, Man, and Cybernetics*, volume 21, pages 660–674.

[Sarwar et al., 2000] Sarwar, B., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Applications of dimensionality reduction in recommender systems – a case study. In *ACM WebKDD 2000 Web Mining for E-Commerce Workshop*.

[Schein et al., 2002] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, Tampere, Finlance. ACM.

[Smith et al., 2001] Smith, M. D., Bailey, J., and Brynjolfsson, E. (2001). Understanding digital markets: Review and assesment. *SSRN eLibrary*.

[Stigler, 2008] Stigler, S. (2008). Fisher and the 5*CHANCE*, 21(4):12 – 12.

[Toffler, 1970] Toffler, A. (1970). *Future Shock*. Random House, New York, NY, USA, 1 edition.

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining - Practical Machine Learning Tools and Techniques*. Second edition. Elsevier, San Francisco, 2nd edition edition.

[Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Zenger et al., 2011] Zenger, R., Eds., Gall, H. C., Bouza, A., and Giger, E. (2011). Collaborative defect prediction: applying collaborative filtering to cross-project defect prediction. Master's thesis, University of Zurich.

[Zhan et al., 2002] Zhan, J., Silvescu, A., and Honavar, V. (2002). Ontology-driven induction of decision trees at multiple levels of abstraction. In *Proceedings of Symposium on Abstraction, Reformulation, and Approximation*, Berlin. Springer-Verlag.

[Zhang et al., 2008] Zhang, L., Xiao, B., Guo, J., and Zhu, C. (2008). A scalable collaborative filtering algorithm based on localized preference. *International Conference on Machine Learning and Cybernetics*, 1:160–167.

[Ziegler, 2005] Ziegler, C.-N. (2005). *Towards Decentralized Recommender Systems*. PhD thesis, Alber-Ludwigs-Universität Freiburg.

[Ziegler and Lausen, 2004] Ziegler, C.-N. and Lausen, G. (2004). Analyzing correlation between trust and user similarity in online communities. In *Proceedings of the 2nd International Conference on Trust Management*, volume 2995, pages 251–265.

Curriculum Vitae

Personal Details

Name:	Amancio Bouza
Place of Birth:	Schaffhausen, Switzerland
Citizenship:	Spain

Education

2008 – 2012:	*Doctor of Science*, Department of Informatics, University of Zurich, Switzerland
2002 – 2007:	*Dipl. Inform.*, University of Zurich, Switzerland

Relevant Research Experience

2007 – 2012:	*Research Assistant*, Dynamic and Distributed Information Systems Group, Department of Informatics, University of Zurich
2007 – 2012:	*Research Assistant*, Software Evolution and Architecture Lab, Department of Informatics, University of Zurich
2011:	*Program Committee Member*, ECML/PKDD 2011 Discovery challenge - Discovery challenge Workshop at the European Conference on Machine Learning and Principles of Knowledge Discovery in Databases
2009:	*Program Committee Member*, International Semantic Web Conference (ISWC) 2009